

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DA PRODUÇÃO E
SISTEMAS – PPGEPS**

ROSEMARY FRANCISCO

**SERVIÇO DE CONTROLE SUPERVISÓRIO: UMA ABORDAGEM DE CONTROLE
PARA PROCESSOS DE NEGÓCIO FLEXÍVEIS**

CURITIBA

2011

ROSEMARY FRANCISCO

**SERVIÇO DE CONTROLE SUPERVISÓRIO: UMA ABORDAGEM DE CONTROLE
PARA PROCESSOS DE NEGÓCIO FLEXÍVEIS**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Produção e Sistemas – PPGEPS – da Pontifícia Universidade Católica do Paraná como requisito parcial para a obtenção do título de Mestre em Engenharia de Produção e Sistemas.

Orientador: Prof. Dr. Eduardo A. Portela Santos
Co-orientador: Prof. Dr. Marco A. B. de Paula

**CURITIBA
2011**

ROSEMARY FRANCISCO

**SERVIÇO DE CONTROLE SUPERVISÓRIO: UMA ABORDAGEM DE CONTROLE
PARA PROCESSOS DE NEGÓCIO FLEXÍVEIS**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Produção e Sistemas da Pontifícia Universidade Católica do Paraná como requisito à obtenção do título de Mestre em Engenharia de Produção e Sistemas.

COMISSÃO EXAMINADORA

Prof. Dr. Eduardo Alves Portela Santos
Orientador

Prof. Dr. Marco Antonio Buseti de Paula
Co-orientador

Prof. Dr. Eduardo de Freitas Rocha Loures
Membro Interno

Prof. Dr. Agnelo Denis Vieira
Membro Interno

Profa. Dra. Maria Beatriz Felgar de Toledo (UNICAMP)
Membro Externo

Prof. Dr. Pedro Manuel Moreira Vaz Antunes de Sousa (IST)
Membro Externo

Curitiba, 28 de Fevereiro de 2011.

Ao meu amado esposo,
minha querida família e
amigos, companheiros
de todas as horas.

AGRADECIMENTOS

Ao meu esposo Marcos pela compreensão durante as minhas ausências e momentos de estudo. Seu bom humor e alegria foram revigorantes nos momentos de cansaço.

À minha família pelo carinho e apoio em todos os momentos e por acreditar no meu potencial.

Ao meu orientador Prof. Dr. Eduardo Alves Portela Santos por acreditar na minha capacidade para o desenvolvimento deste trabalho e me auxiliar em todas as etapas da pesquisa.

Ao meu co-orientador Prof. Dr. Marco Antonio Buseti de Paula pela atenção despendida durante as reuniões de planejamento para o desenvolvimento deste trabalho.

Ao Prof. Dr. Agnelo Denis Vieira pelo auxílio com os aplicativos de SCT.

Ao Prof. Dr. Eduardo de Freitas Rocha Loures por me enveredar no caminho da pesquisa.

Ao Prof. Dr. Mehran Misaghi, por acompanhar os meus passos desde a graduação. Seu incentivo e atenção são sempre motivadores.

Aos meus amigos e companheiros de tutoria que forneceram dicas essenciais e colaboraram para o desenvolvimento desta pesquisa.

E a todos que contribuíram para que a conclusão deste trabalho fosse possível.

*“Quem sobrevive não é
o mais forte ou o mais inteligente,
e sim aquele que melhor se
adapta às mudanças.”*

Charles Darwin

RESUMO

Para se tornarem ou se manterem competitivas, as empresas estão fazendo uso efetivo de tecnologia da informação visando o suporte aos processos de negócio. Os Sistemas de Informação baseados em Processos (PAIS) estão sendo utilizados com o objetivo de oferecer este tipo de suporte aos processos, possibilitando, desta forma, um melhor desempenho e consistência das informações. Quando um processo é modelado são considerados: os objetivos da organização, sua infraestrutura, o contexto, restrições, entre outros pontos que podem impactar no resultado do processo. No entanto, quando este modelo de processo é implantado e executado em um PAIS, esta visão idealizada é muitas vezes quebrada. Com base neste cenário, as empresas estão demandando uma maior flexibilidade dos seus PAIS. Entretanto, o incremento de flexibilidade nos PAIS usualmente está associado a um menor grau de suporte ao usuário de tais sistemas, conseqüentemente exigindo usuários mais experientes para a execução das atividades dos processos. Por outro lado, o excesso de suporte pode restringir a flexibilidade dos processos e prejudicar o seu desempenho, visto que decisões estratégicas momentâneas não poderão ser realizadas em um ambiente restrito e com pouca flexibilidade. Desta forma, para prover um PAIS flexível com elevado grau de suporte, uma plataforma inteligente de apoio a decisão torna-se necessária. Neste trabalho é proposto um Serviço de Controle Supervisório (SCS) que, quando usado em combinação com um PAIS flexível, dá suporte aos usuários durante a execução das atividades. Esse suporte é implementado por meio da aplicação das regras de negócio durante a execução das atividades do processo; fazendo com que os usuários tenham conhecimento a respeito das regras de negócio aplicadas às atividades, e portanto, executem somente as atividades habilitadas ou, dependendo do seu perfil no processo, executem as atividades que possibilitariam o melhor resultado para o alcance das metas estipuladas para os processos. Tal aplicação das regras de negócio acontece de forma contínua, de acordo com a seqüência de eventos observadas na execução dos processos no PAIS.

Palavras-chave: Sistemas de Controle e Supervisão. Business Process Management (BPM). Process-Aware Information System (PAIS).

ABSTRACT

In order to become or remain competitive, companies are making effective use of information technology aiming at support to the business processes. Process-Aware Information Systems (PAIS) are being used in order to offer this type of process support, allowing thus a better performance and consistency of information. When a process is modeled is considered: the goals of the organization, its infrastructure, the context, restrictions, among other points that can impact in the result of process execution. However, when this process model is deployed and executed in a PAIS, this idealized vision often is broken. Based on this context, companies are demanding a greater flexibility of its PAIS. However, the increase of flexibility in a PAIS usually is associated with a lesser degree of user support, therefore requiring more experienced users to execute the process activities. Moreover, excessive support may restrict the flexibility of processes and harm its performance, whereas momentary strategic decisions cannot be performed in a restricted environment with little flexibility. Thus, to provide a flexible PAIS with high degree of support, an intelligent platform to support the decision becomes necessary. This research proposes a Supervisory Control Service (SCS) which, when used in combination with a flexible PAIS, provides support to end users during the execution of activities. This support is implemented by the application of business rules during the execution of process activities, making that the users have knowledge regarding the applied business rules to the activities, and therefore execute only authorized activities or, depending on your profile in the process, execute activities in order to enable the best result for the reach of the goals defined for the processes. This application of the business rules happens continuously, according to the sequence of events observed in the execution of the process activities in a PAIS.

Key-words: Supervisory Control Systems. Business Process Management (BPM). Process-Aware Information System (PAIS).

LISTA DE FIGURAS

Figura 1.1 - Etapas da Pesquisa	25
Figura 2.1 - Visão Macro de um Processo	32
Figura 2.2 - Ciclo de vida do BPM.....	35
Figura 2.3 - Ciclo de vida e principais elementos e ferramentas do BPM	36
Figura 2.4– Perspectivas essenciais para a modelagem de Processos de Negócio.	37
Figura 2.5 - Classificação de recursos	40
Figura 2.6–Relação PAIS, BPM e WFM.....	47
Figura 2.7–Arquitetura WFMS.....	48
Figura 2.8–Cenário dos tipos de PAIS	53
Figura 2.9 - Trajetória típica de um sistema a eventos discretos	56
Figura 2.10 - Exemplo de um autômato determinístico	59
Figura 2.11 - Autômato que marca a linguagem consistindo de todas as palavras de a e b	60
Figura 2.12 - Autômato não-acessível e não co-acessível	61
Figura 2.13 - Autômato acessível.....	61
Figura 2.14 - Autômato Trim.....	62
Figura 2.15 – Autômatos G1e G2.....	63
Figura 2.16 - Autômato G1 G2	64
Figura 2.17 - Esquema do controle monolítico	67
Figura 2.18 - Esquema de controle modular local	69
Figura 3.1 – Modelo para representação de uma atividade constituinte de um processo de negócio	73
Figura 3.2–Visão Geral da Arquitetura de Controle Supervisório.....	74
Figura 3.3 - Ciclo de Desenvolvimento de Controle e Supervisão	77
Figura 3.4 - Arquitetura do Sistema de Controle e Supervisão	78
Figura 4.1–Visão geral das técnicas de mineração no ProM	82
Figura 4.2 - Visão Geral dos provedores do OS no ProM	84
Figura 4.3 - Arquitetura do SCS	85
Figura 4.4 - Diagrama de Classes do SCS.....	87
Figura 4.5 - Diagrama de Classes OS.....	88
Figura 4.6 - Comunicação entre o PAIS, OS e SCS	89
Figura 4.7 - Comunicação entre os níveis do SCS – Recebimento de Solicitação ...	90

Figura 4.8 - Comunicação entre os níveis do SCS - Retorno da Execução	91
Figura 4.9 - Diagrama de Classes do componente CMPAutomata	92
Figura 5.1 - Representação dos Autômatos das Atividades.....	98
Figura 5.2 - Representação das especificações de controle	99
Figura 5.3 - Representação das Plantas Locais.....	100
Figura 5.4 - Representação dos Supervisores Locais.....	101
Figura 5.5 - Representação dos Supervisores Locais Reduzidos.....	102
Figura 5.6 – Widget com o guia do SCS	104
Figura 5.7 – Visão Geral da Arquitetura do SCS.....	106
Figura 5.8 – Estrutura do arquivo RulesCMLevel.xml	107
Figura 5.9 - Especificação das regras no SCS.....	108
Figura 5.10–Atividades habilitadas para execução inicialmente no Bizagi.....	111
Figura 5.11–Início da execução da atividade B no Bizagi	112
Figura 5.12–Execução do SCS com base na execução da atividade B no Bizagi ...	112
Figura 5.13–Atividades habilitadas para execução durante a execução da atividade B no Bizagi	113
Figura 5.14–Execução do SCS após a finalização da execução da atividade B no Bizagi.....	113
Figura 5.15–Atividades habilitadas para execução após a finalização da execução da atividade B no Bizagi.....	114
Figura 5.16–Atividades habilitadas para execução inicialmente no YAWL	115
Figura 5.17–Início da execução da atividade B no YAWL.....	115
Figura 5.18–Execução do SCS com base na execução da atividade B no YAWL..	116
Figura 5.19–Finalização da execução da atividade B no YAWL	117
Figura 5.20–Execução do SCS após a finalização da execução da atividade B no YAWL	117
Figura 5.21–Atividades habilitadas para execução após a finalização da execução da atividade B.....	118
Figura A.1– Modelo do SGA no Bizagi.....	134
Figura A.2– Perfis do Processo no Bizagi	134
Figura A.3– Criação de usuário no Bizagi	135
Figura A.4– Atribuição de perfil para o usuário no Bizagi.....	136
Figura A.5– Expressão Consulta SCS criada no Bizagi	137
Figura A.6– Propriedades da Consulta SCS	137

Figura A.7– Configuração da localização do Webservice no Bizagi	138
Figura A.8– Métodos do Webservice apresentados no Bizagi.....	139
Figura A.9– Configuração do valor do parâmetro do método do Webservice no Bizagi.....	139
Figura A.10– Configuração dos eventos das atividades do processo no Bizagi	140
Figura A.11– UserField criado para apresentar o Widget no Bizagi	140
Figura A.12– Código utilizado no UserField para a configuração do Widget.....	141
Figura A.13– Selecionado o UserField para a configuração do Widget no formulário das atividades	141
Figura A.14– UserField configurado no formulário das atividades	142
Figura B.15– Modelo do SGA no YAWL.....	143
Figura B.16– Configurando a interface personalizada no YAWL	143
Figura B.17– Perfis do Processo no YAWL.....	144
Figura B.18– Criação de usuário no YAWL.....	145
Figura B.19– Configuração do Webservice no YAWL	145
Figura B.20– Webservice configurado no YAWL.....	146
Figura C.21– Implementação da Classe SupervisoryControlProvider.....	147
Figura D.22– WSDL do Webservice	148

LISTA DE TABELAS

Tabela 2.1–Principais elementos dos Processos	32
Tabela 2.2 - Tipos de Processos.....	33
Tabela 3.1 – Semântica de estados e eventos empregados no modelo da Figura 3.1	73
Tabela 5.1–Relação entre as atividades do SGA e restrições de controle.....	96
Tabela E.1–Plano de testes elaborado para a condução do experimento	149
Tabela E.2–Caso de testes 01 elaborado para a condução do experimento	150
Tabela E.3–Caso de testes 02 elaborado para a condução do experimento	150
Tabela E.4–Caso de testes 03 elaborado para a condução do experimento	151

LISTA DE ABREVIATURAS E SIGLAS

AF	– Autômato Finito
AFD	– Autômato Finito Determinístico
A2A	– Aplicação-Aplicação
BAM	– Business Activity Monitoring
BPA	– Business Process Analysis
BPD	– Business Process Diagram
BPEL	– Business Process Execution Language
BPM	– Business Process Management
BPMI	– Business Process Management Initiative
BPMN	– Business Process Management Notation
BPMS	– Business Process Management Systems
BRE	– Business Rules Engine
CASE	– Computer-Aided Software Engineering
CH	– Case Handling
CL	– Camada de Comunicação
CRM	– Customer Relationship Management
EAI	– Enterprise Application Integration
EPC	– Event Driven Process Chain
JEE	– Java Enterprise Edition
LMC	– Local Modular Control
MS	– Supervisores Modulares
OS	– Operational Support
PAIS	– Process-Aware Information Software
PS	– Sistema Produto
PROM	– Process Mining Framework
P2A	– Pessoa-Aplicação
P2P	– Pessoa-Pessoa
RdP	– Redes de Petri
RSP	– Representação Sistema Produto
SCA	– Arquitetura de Controle e Supervisão

SCS	– Serviço de Controle Supervisório
SCSA	– Arquitetura de Serviço de Controle Supervisório
SCT	– Teoria de Controle de Supervisório
SED	– Sistemas a Eventos Discretos
SEDC	– Sistemas a Eventos Discretos Controláveis
SEDO	– Sistemas a Eventos Discretos Observáveis
SGA	– Sistema de Gerenciamento de Anomalias
SGBD	– Sistema Gerenciador de Banco de Dados
SOA	– Service-Oriented Architecture
SOAP	– Simple Object Access Protocol
TI	– Tecnologia da Informação
UML	– Unified Modeling Language
XML	– eXtensible Markup Language
WFM	– Workflow Management
WFMS	– Workflow Management Systems
W3C	– World Wide Web Consortium
WSDL	– Web Services Description Language
YAWL	– Yet Another Workflow Language

SUMÁRIO

1	INTRODUÇÃO	17
1.1	PROBLEMA DE PESQUISA.....	19
1.2	OBJETIVOS	20
1.2.1	Objetivos específicos	21
1.3	JUSTIFICATIVA	22
1.4	METODOLOGIA	23
1.4.1	Procedimentos metodológicos	24
1.4.2	Limitação da pesquisa	26
1.5	ESTRUTURA DA DISSERTAÇÃO.....	26
2	FUNDAMENTOS.....	29
2.1	GERENCIAMENTO DE PROCESSOS DE NEGÓCIO - BPM	29
2.1.1	Definição de processos.....	31
2.1.2	Ciclo de vida e elementos do BPM	35
2.1.3	Sistemas de gestão de processos – BPMS	43
2.1.4	Arquitetura Orientada a Serviço – SOA	44
2.2	PROCESS-AWARE INFORMATION SYSTEMS – PAIS	46
2.2.1	Tipos de PAIS	50
2.2.2	Flexibilidade no PAIS	52
2.3	TEORIA DE CONTROLE DE SUPERVISÓRIO.....	54
2.3.1	Sistemas a eventos discretos – SED	55
2.3.2	Linguagens Formais	56
2.3.3	Operações sobre as linguagens	57
2.3.4	Teoria de autômatos	58
2.3.5	Linguagens geradas e marcadas.....	59
2.3.6	Operações sobre autômatos.....	60
2.3.7	Supervisão centralizada (Controle Monolítico).....	64
2.3.8	Abordagem Modular Local da Teoria de Controle Supervisório.....	68
2.4	CONCLUSÃO DO CAPÍTULO	70
3	ABORDAGEM DE CONTROLE UTILIZADA NO SCS	72
3.1	VISÃO GERAL DO SCS	72
3.2	CICLO DE DESENVOLVIMENTO DE CONTROLE E SUPERVISÃO ...	76

3.3	FORMALIZANDO A ARQUITETURA DE CONTROLE.....	78
3.4	CONCLUSÃO DO CAPÍTULO	79
4	PROJETO DO SCS.....	81
4.1	PROM FRAMEWORK.....	81
4.2	ESPECIFICAÇÃO E IMPLEMENTAÇÃO DO SCS NO PROM.....	83
4.2.1	ARQUITETURA DO SCS.....	84
4.2.2	CMPAutomata.....	92
4.3	CONCLUSÃO DO CAPÍTULO	93
5	CONFIGURAÇÃO E EXECUÇÃO DO SCS.....	94
5.1	SISTEMA DE GERENCIAMENTO DE ANOMALIAS – SGA	94
5.2	RECURSOS PARA INTEGRAÇÃO DO PAIS COM O SCS.....	103
5.3	CONFIGURAÇÃO DO SCS PARA SUPERVISIONAR UM PAIS	106
5.4	CONFIGURAÇÃO DO PAIS	109
5.5	PLANEJAMENTO E EXECUÇÃO DO SGA.....	109
5.5.1	Execução do SCS no Bizagi	110
5.5.2	Execução do SCS no YAWL.....	114
5.6	CONCLUSÃO DO CAPÍTULO	118
6	CONCLUSÕES	120
6.1	AVALIAÇÃO DE RESULTADOS.....	121
6.2	SUGESTÃO PARA TRABALHOS FUTUROS.....	123
	REFERÊNCIAS.....	125
	APÊNDICES	133
	APÊNDICE A – CONFIGURAÇÃO DO PAIS BIZAGI	134
	APÊNDICE B – CONFIGURAÇÃO DO PAIS YAWL.....	143
	APÊNDICE C – IMPLEMENTAÇÃO SUPERVISORYCONTROLPROVIDER....	147
	APÊNDICE D – IMPLEMENTAÇÃO DO WEBSERVICE	148
	APÊNDICE E – ESPECIFICAÇÃO DO PLANO E CASOS DE TESTE.....	149

1 INTRODUÇÃO

Em uma economia globalizada, as empresas necessitam cada vez mais de agilidade nos seus processos, de informações e conhecimento disponíveis e facilmente acessíveis. Mais ainda, é preciso que tal conhecimento auxilie de forma inteligente a gestão da organização, para que esta consiga manter-se no mercado de forma competitiva (MAGALHÃES et al. 2004). Atualmente há um consenso de que o sucesso econômico de uma empresa depende da sua capacidade para reagir às mudanças no seu ambiente, de forma rápida e flexível (WEBER et al. 2008). A contínua transformação do negócio - onde a mudança é a norma e não a exceção - é necessária para manter a competitividade e proporcionar produtos e serviços feitos sob medida (CARLSEN et al. 1997). Um dos grandes desafios enfrentados pelas organizações no ambiente de hoje é transformar idéias e conceitos em produtos e serviços em um ritmo sempre crescente (DUMAS et al. 2005). O aumento da concorrência global está levando as empresas a reduzir o tempo de resposta ao lançamento de novos produtos e de oferecer preços competitivos. Diversidade, as flutuações na demanda, o curto ciclo de vida dos produtos devido à introdução freqüente de novas necessidades, além do aumento das expectativas do cliente em termos de qualidade e tempo de entrega, são hoje alguns dos principais desafios com que as empresas têm de lidar de forma a manter a competitividade e permanência no mercado. Por estas razões as empresas têm reconhecido a agilidade dos negócios como uma vantagem competitiva, que é fundamental para ser capaz de lidar com as tendências de negócios mencionados.

Neste cenário, o uso efetivo de tecnologia da informação visando o suporte aos processos de negócio tem sido largamente discutido. Muitas idéias têm surgido, dentre elas o alinhamento das estratégias de tecnologia com a arquitetura para desenho, execução, monitoramento e gerência dos processos de negócio, criação de parcerias entre os gestores dos processos de negócio e a área de tecnologia, descoberta de vários caminhos de sucesso com a integração de tecnologia aos processos de negócio, dentre outros. Para tanto, é preciso que os processos de negócio estejam alinhados às estratégias e metas da organização, e que o planejamento de tecnologia seja coerente e integrado a tais processos. Somente

desta forma torna-se possível disponibilizar informações de qualidade, em tempo hábil, para o nível estratégico da organização (MAGALHÃES et al. 2004).

Para que este alinhamento entre negócio e tecnologia seja possível, sistemas de informação para o suporte aos processos são requisitos para possibilitar que as empresas ajustem ou regulem rapidamente seus processos de negócio de acordo com as características do seu ambiente (SCHONENBERG et al. 2008c). Segundo Weber et al. (2008), os sistemas de informação que possibilitam o gerenciamento de processos de negócio – Process-Aware Information Systems (PAIS), oferecem perspectivas promissoras a esse respeito, e um crescente interesse em alinhar os sistemas de informação com orientação para o processo, pode ser observado. Process-Aware Information System (PAIS) é definido como um sistema de software que gerencia e executa processos operacionais envolvendo pessoas, aplicações e / ou fontes de informação sobre os fundamentos dos modelos de processo (DUMAS et al. 2005). Exemplos de PAIS são os Workflow Management Systems (WFMSs), como Staffware e Peoplesoft; os sistemas Case Handling (CH), como FLOWer; os sistemas de Customer Relationship Management (CRM), como os sistemas CRM da Oracle, SAP ou Salesforce.com, dentre outros (DONGEN et al. 2005).

Neste contexto, as empresas estão exigindo uma maior flexibilidade de seus PAIS, pois rápidas mudanças no ambiente de negócio exigem que os sistemas de informação de suporte sejam flexíveis. De acordo com Schonenberg et al. (2008b), um elemento-chave de processos flexíveis é a sua capacidade de lidar com as mudanças previstas e não previstas no contexto ou ambiente em que operam. Neste sentido, a flexibilidade de um processo reflete a sua capacidade de lidar com tais mudanças, através da variação ou adaptação na execução das atividades, embora mantendo o formato essencial, a contínua aplicação das regras de negócio por meio destes sistemas de suporte e gerenciamento do processo. Embora o conceito de flexibilidade seja relativamente simples, sua implementação é mais difícil de conseguir na prática (SCHONENBERG et al. 2008a) (SCHONENBERG et al. 2008b). Em geral, durante a execução de um processo de negócio, em um PAIS flexível, o usuário tem a opção de selecionar qual atividade realizar dentre as diversas atividades que estão habilitadas para o cenário atual de execução do processo de negócio (SCHONENBERG et al. 2008c). No entanto, os sistemas de informação que permitem flexibilidade têm uma tendência à falta de suporte ao processo relacionado. Para todas as abordagens de flexibilidade existentes, o

apoio ao usuário fornecido pelo PAIS diminui com o aumento da flexibilidade, uma vez que estão disponíveis outras opções, que exigem que os usuários tenham um conhecimento aprofundado sobre os processos que estão trabalhando. Tradicionalmente, este problema é resolvido por meio da capacitação dos usuários, por exemplo, tornando-os mais conscientes do contexto em que um processo é executado, ou por meio da introdução de restrições no PAIS que permitam conduzir os usuários; neste último caso, porém, sacrificando a flexibilidade. É relativamente fácil desenvolver sistemas que são extremamente flexíveis, no entanto estes irão fornecer pouco ou nenhum suporte para os usuários (AALST et al. 2009). Ambas as opções, no entanto, não são satisfatórias e limitam a aplicação prática do PAIS flexível. Como resultado, estes sistemas não são utilizados para suportar dinamicamente as mudanças dos processos de negócio ou os processos são suportados de uma maneira restrita, por exemplo, mudanças não são permitidas ou são manipuladas fora do sistema, sem o suporte do sistema (AALST et al. 2009).

Várias abordagens são propostas para resolver a dicotomia entre suporte e flexibilidade. Algumas destas abordagens tentam evitar a mudança, por exemplo, por meio da geração de caminhos alternativos (AALST et al. 2005; AGOSTINI e MICHELIS 2000), ou adiando a seleção do comportamento desejado (ADAMS et al. 2006). Outras abordagens permitem a alteração do modelo do processo para uma única instância do processo, ou para todas as instâncias na fase de modelagem ou mesmo na fase de execução (ELLIS et al. 1995; RINDERLE et al. 2004; WESKE 2001). Outras abordagens utilizam restrições como item de controle entre o suporte e a flexibilidade durante a execução dos processos (PESIC 2008; WAINER e DE LIMA BEZERRA 2003).

O estudo destas abordagens demonstrou que ainda há uma limitação no suporte aos PAISs flexíveis e, desta forma, o presente trabalho tem como objetivo propor uma nova abordagem que permita lidar com a flexibilidade no PAIS, mantendo, no entanto, o apoio aos usuários durante a execução das atividades dos processos de negócio.

1.1 PROBLEMA DE PESQUISA

Com base nos estudos realizados, quando um PAIS é desenvolvido e implementado para oferecer flexibilidade no gerenciamento dos processos do negócio, muitas vezes o suporte aos usuários, no momento de execução do processo, é preterido. O contrário também ocorre, para que um PAIS ofereça suporte aos usuários durante a execução dos processos de negócio, este deve ser desenvolvido e implementado de forma estruturada a fim de garantir este suporte, no entanto, nestes casos, a flexibilidade na implementação de novos requisitos e/ou de modificações necessárias para atender às constantes demandas dos processos é reduzida. Portanto, quanto maior a flexibilidade empregada no PAIS, a fim de permitir uma maior liberdade na tomada de decisão para a execução do processo, menor será o suporte empregado. Isto significa que a flexibilidade no PAIS demanda usuários qualificados, conhecedores da estratégia da empresa, contexto e objetivo do processo.

Além disso, a tomada de decisão na execução dos processos envolve a construção de conhecimento. Quando um modelo de processo é elaborado, são considerados os objetivos da organização, sua infra-estrutura, o contexto, restrições, recursos disponíveis entre outros. No entanto, quando este modelo de processo é implantado e executado, esta visão idealizada é muitas vezes quebrada, em virtude das demandas repentinas e em alguns casos, específicas, que ocorrem no ambiente dos negócios. Em particular, modelos de processos geralmente assumem que as atividades definidas devem acontecer dentro de um determinado contexto. Quando tais pressupostos não são cumpridos, os usuários devem tomar decisões sobre formas alternativas para alcançar a meta e ainda concluir o processo de acordo com as regras de negócio estipuladas.

Neste contexto, surge o seguinte problema para pesquisa:

Como manter a flexibilidade do PAIS e ainda permitir o suporte à tomada de decisão aos usuários durante a execução dos seus processos de negócio?

1.2 OBJETIVOS

A presente pesquisa tem como objetivo geral propor um serviço de controle e supervisão que possibilite manter a flexibilidade do PAIS e também permitir o suporte aos usuários durante a execução dos processos de negócio suportados por um sistema de informação (PAIS).

Com base nos eventos que ocorrem no ambiente do processo, o serviço de controle retornará uma lista de possíveis passos (atividades) que podem ou devem ser executados pelos usuários, de acordo com o perfil do usuário no processo, a fim de garantir que as regras de negócio preestabelecidas, não sejam violadas. Nos casos em que o usuário possui um perfil de operador das atividades do processo, o serviço poderá ser configurado para obrigar a execução das atividades de acordo com as regras definidas, garantindo desta forma, a execução do processo segundo o modelo prescrito. Porém, caso o usuário tenha um perfil de analista do processo, este possuirá privilégios para alterar as regras de negócio definidas no serviço ou mesmo ignorá-las para uma instância de processo específica, a fim de permitir a flexibilidade para a execução das atividades do processo.

1.2.1 Objetivos específicos

Para atender o objetivo geral da presente pesquisa, foram definidos os seguintes objetivos específicos:

- a) pesquisar e revisar os principais conceitos relacionados à pesquisa: Business Process Management (BPM), Process-Aware Information System (PAIS) e Teoria de Controle de Supervisório a fim de identificar as melhores práticas e também limitações que servirão como base na definição da abordagem proposta;
- b) projetar e definir a estrutura do serviço de controle de supervisório para possibilitar o monitoramento e controle do processo e suas regras de negócio;
- c) implementar o serviço de controle de supervisório projetado de acordo com a estrutura definida;
- d) implementar os recursos que permitirão a integração do serviço de controle de supervisório com as ferramentas PAISs, possibilitando desta forma que

- o serviço monitore os eventos e retorne o suporte necessário para os usuários durante a execução do processo;
- e) elaborar o plano e os casos de testes para a integração do serviço de controle e supervisão em ferramentas PAIS a fim de demonstrar a aplicação da abordagem proposta pela presente pesquisa.

1.3 JUSTIFICATIVA

Problemas freqüentes das organizações acarretam em falta de controle e altos custos para adaptações às mudanças do negócio, e para a implantação de inovações e novas soluções que são vitais para atender à volatilidade do mercado. Dentre estes problemas pode-se citar o conhecimento superficial dos seus processos de trabalho, a falta de agilidade na reformulação dos mesmos, a fraca aderência das aplicações de TI aos processos de negócio da organização, a redundância de informações, além da diversidade de tecnologias disponíveis para dar suporte às atividades da organização, desde o nível operacional até o nível estratégico (MAGALHÃES et al. 2004).

Outra característica que deve ser considerada se refere ao comportamento das pessoas envolvidas nos processos, frente a dificuldades encontradas no seu dia-a-dia. As pessoas normalmente mudam seu comportamento habitual quando confrontados com um prazo. Isto significa que muitos usuários podem ignorar algumas tarefas ou exigir menos informação durante a execução de um processo em um PAIS, quando o custo de perder o prazo é maior do que o custo ou a perda da qualidade de serviço (AALST, ROSEMAN e DUMAS 2007). Isto significa que regras de negócio preestabelecidas com base no nível estratégico da empresa, podem ser violadas para que o prazo ou ainda a demanda específica seja cumprida. Porém, para tomar este tipo de decisão é necessário que os usuários possuam um conhecimento abrangente em relação ao processo em que está envolvido e a relação deste com os demais processos da organização. Caso contrário, é possível que estes usuários tomem decisões inadequadas, utilizando como parâmetro para sua tomada de decisão apenas o seu ambiente de trabalho.

Portanto, possuir flexibilidade na tomada de decisão é essencial em muitas situações durante a execução de um processo, porém, esta raramente é apoiada e suportada de forma consistente pelos PAIS. Desta forma, possuir um serviço que permita às organizações capacidade para reagir às mudanças no seu ambiente de uma forma rápida e flexível é objetivo deste trabalho. Este serviço irá apoiar os usuários na execução dos processos a fim de suportá-los e auxiliá-los na tomada de decisão.

1.4 METODOLOGIA

A metodologia de pesquisa pode ser definida como o conjunto de métodos e técnicas utilizados para a realização de uma pesquisa. De acordo com Paim et al. (2009), a metodologia de trabalho de uma pesquisa tem como principal objetivo conferir a validade ao conhecimento científico produzido. Thiollent (1998) descreve que a metodologia de pesquisa consiste em analisar as características dos vários métodos disponíveis, avaliar suas capacidades, potencialidades, limitações ou distorções e criticar os pressupostos ou as implicações de sua utilização. Thiollent (1998) enfatiza que além de ser uma disciplina que estuda os métodos, a metodologia é também considerada como o modo de conduzir a pesquisa.

Por meio da metodologia de pesquisa é possível planejar e estruturar o desenvolvimento de uma pesquisa, de maneira sistematizada, com base em uma investigação científica a respeito de um fenômeno observado na vida real. Para o desenvolvimento de uma pesquisa é possível a utilização de um ou vários métodos combinados de observação, de maneira a apreender fatos e dados dessa realidade, com a intenção de entender, explicar e, se possível ou necessário, aplicá-lo ou replicá-lo em favor de outros eventos ou situações semelhantes (BERTO e NAKANO1998).

De acordo com Gil (2002), a pesquisa aplicada objetiva gerar conhecimentos para a aplicação prática dirigida à solução de problemas específicos. A pesquisa qualitativa considera que há uma relação dinâmica entre o mundo real e o sujeito, isto é, um vínculo indissociável entre o mundo objetivo e a subjetividade do sujeito que não pode ser traduzido em números. A interpretação dos fenômenos e a

atribuição de significados são básicos no processo de pesquisa qualitativa. Sua preocupação central (GIL 2002) tem foco na identificação dos fatores que determinam ou que contribuem para a ocorrência de fenômenos. Este tipo de pesquisa além de possuir caráter exploratório e descritivo, utiliza um estudo sistemático que permitirá a avaliação dos fenômenos investigados.

Com o objetivo de conduzir e conferir a validade do presente trabalho utilizou-se os métodos de pesquisa: sob o ponto de vista da natureza do trabalho, a pesquisa aplicada visto que a proposta é uma nova abordagem para o problema da flexibilidade que ocorre nos processos de negócio que são suportados pelos PAISs; sob o ponto de vista da abordagem, utilizou-se a pesquisa qualitativa, visto que a avaliação dos resultados se dá por meio da observação da aplicação da abordagem proposta, ou seja, não requer o uso de métodos e técnicas estatísticas; e por fim, sob o ponto de vista de vista dos objetivos, utilizou-se a pesquisa explicativa, visto que este tipo de pesquisa permite aprofundar o conhecimento da realidade, além de identificar o cenário produzido por meio da aplicação da abordagem proposta em um ambiente simulado.

Quanto aos procedimentos técnicos, utilizou-se a pesquisa experimental. Segundo Gil (2002), a pesquisa experimental consiste em determinar um objeto de estudo, selecionar as variáveis que seriam capazes de influenciá-lo, definir as formas de controle e de observação dos efeitos que a variável produz no objeto. De acordo com Calegare (2001), o foco dos experimentos é a descoberta, o rumo ao desconhecido, para aperfeiçoamento do processo ou otimização de suas saídas.

Com base nestes métodos e procedimentos, elaborou-se um estudo experimental, pelo qual se objetiva implementar, integrar e avaliar a aplicabilidade do serviço de controle e supervisão proposto pelo presente trabalho.

1.4.1 Procedimentos metodológicos

O desenvolvimento da presente pesquisa compreendeu a realização das etapas ilustradas na figura 1.1:

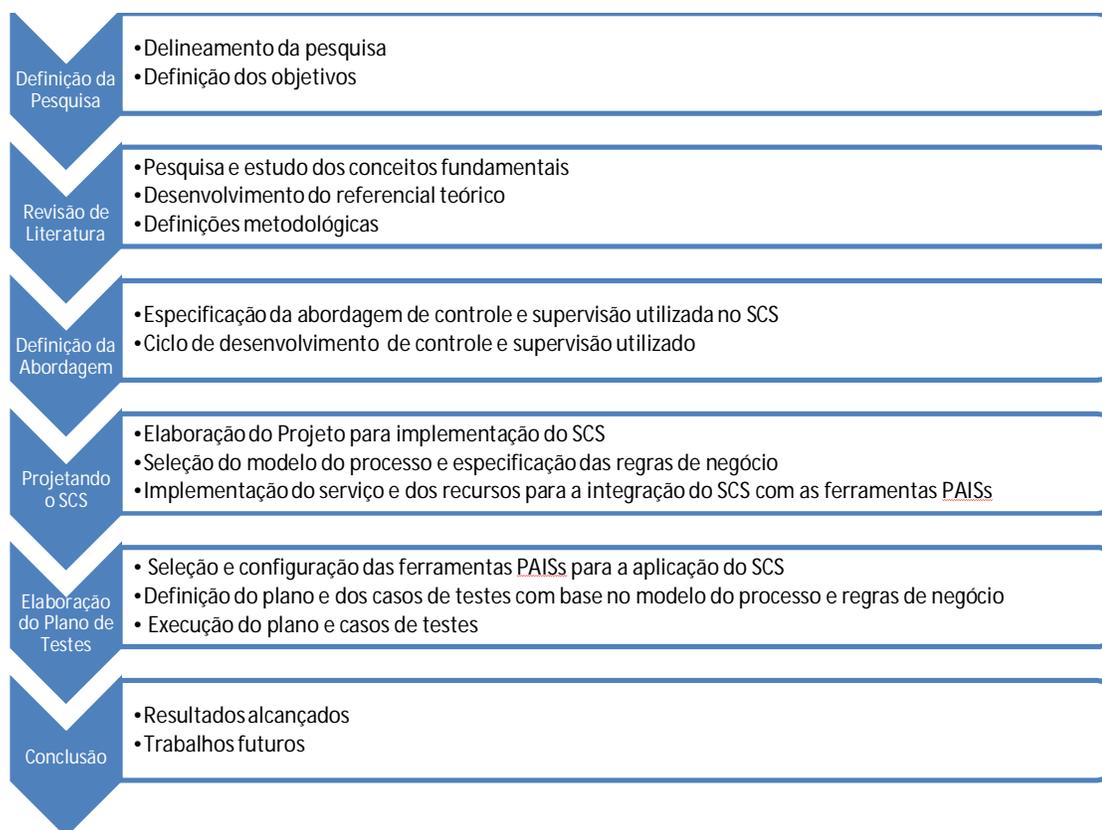


Figura 1.1 - Etapas da Pesquisa

Na primeira etapa da pesquisa foi identificado o problema de pesquisa e a delimitação do tema, a justificativa, o objetivo geral e os objetivos específicos. Na segunda etapa, foi realizada uma ampla revisão de literatura acerca dos fundamentos necessários para o desenvolvimento da pesquisa; os resultados desta etapa foram transformados em uma base para a terceira etapa da pesquisa: a definição da abordagem para a implementação do serviço de controle e supervisão proposto para solucionar o problema de pesquisa identificado. Na terceira etapa também foi definido o ciclo de desenvolvimento de controle e supervisão que seria utilizado. Na quarta etapa da pesquisa foi elaborado o projeto do SCS contendo toda a especificação necessária para a implementação do serviço e recursos para integração com as ferramentas PAIS, além da seleção do modelo do processo e a especificação das regras de negócio que seriam aplicadas durante a execução do processo. Na quinta etapa da pesquisa foram selecionadas as ferramentas PAIS que seriam utilizadas no experimento e elaborado o plano de testes que permitiu a aplicação do serviço implementado nas ferramentas PAISs selecionadas. Além disso, foram definidos os casos de testes com a especificação dos procedimentos

necessários para a condução do experimento e avaliação da aplicação do serviço proposto. Por fim, na sexta e última etapa, foram avaliados os resultados alcançados e discutidas as possíveis continuidades da pesquisa.

1.4.2 Limitação da pesquisa

A utilização da pesquisa experimental de laboratório pode ser considerada um fator limitante, pois “no laboratório pode-se alcançar um alto grau de exatidão, razão pela qual a medição e os problemas de controle são geralmente mais simples que os encontrados em campo” (MOREIRA 2002). Em situações reais, há sempre ocorrência de imprevistos e outros fatores que podem influenciar na condução e resultados da pesquisa.

A pesquisa experimental é considerada o melhor exemplo de pesquisa científica, pois há um alto nível de controle da situação, podem-se isolar todas as estruturas de qualquer interferência do meio exterior, gerando maior confiabilidade em seus resultados. Mesmo assim ela é flexível, podendo dar inúmeras respostas diferentes a problemas diferentes com um único experimento (KERLINGER 1979).

Esta característica de ambiente controlado permite classificar a pesquisa experimental como um método tanto positivo quanto negativo. Positivo devido à maior credibilidade já que a interferência do meio é praticamente nula, tendo o pesquisador total autonomia sobre seu objeto de estudo; negativa, pois, já que tirando o objeto de seu meio natural as análises serão parciais e não aplicáveis nas relações fora do ambiente propiciado pelo pesquisador (KERLINGER 1979).

Portanto, diante dos resultados alcançados ao final da pesquisa, sugere-se a continuidade desta pesquisa com a aplicação do serviço em ambiente não controlado.

1.5 ESTRUTURA DA DISSERTAÇÃO

Baseado nas etapas da pesquisa, a dissertação também foi organizada em seis capítulos. No primeiro capítulo, no qual esta seção faz parte, o tema e o problema de pesquisa são apresentados. A justificativa, o objetivo geral, os objetivos específicos e a abordagem metodológica utilizada para o desenvolvimento da pesquisa são descritos neste capítulo.

No segundo capítulo são apresentados os conceitos necessários para a realização do trabalho. Neste capítulo são discutidos os principais conceitos de gestão de processos de negócio; são apresentadas as principais características dos sistemas de informação que suportam os processos de negócio, os Process-Aware Information Systems (PAIS); além da conceituação da teoria de controle de supervisão e seus principais elementos.

O terceiro capítulo descreve a abordagem de controle utilizada para o monitoramento e supervisão dos processos proposto pelo presente trabalho. Este capítulo utiliza como base os conceitos de teoria de controle de supervisão fundamentados no capítulo 2. Como resultado deste capítulo, obtém-se a definição detalhada da abordagem de controle e supervisão elaborada para a solução do problema de pesquisa proposto, além do ciclo de desenvolvimento utilizado para o controle e supervisão.

No quarto capítulo é apresentado com detalhes o projeto para a implementação da abordagem proposta. Neste capítulo é apresentada a estrutura e tecnologias utilizadas para a implementação do serviço - Supervisory Control Service (SCS), bem como os modelos e diagramas UML – Unified Modeling Language – gerados para a construção do serviço de controle e supervisão.

O quinto capítulo apresenta o modelo de processo e as restrições – regras de negócio – utilizadas para a especificação das regras de controle para a condução do experimento com o SCS. Neste capítulo também são apresentados o plano e os casos de testes que foram elaborados para a condução do experimento, além da demonstração do experimento realizado. Por meio do plano e casos de testes foi possível especificar os parâmetros, as configurações e os procedimentos utilizados para a integração do serviço bem como descrever os detalhes da aplicação do serviço nas duas ferramentas PAIS selecionadas.

Por fim, no sexto e último capítulo, são discutidos os resultados alcançados, a análise dos objetivos propostos e a comparação destes objetivos com os resultados

alcançados. Além disso, são discutidas as principais contribuições do trabalho apresentado e sugestões para trabalhos futuros.

2 FUNDAMENTOS

Neste capítulo são apresentados os fundamentos e formalismos utilizados no desenvolvimento desta pesquisa. O capítulo inicia com uma visão geral do gerenciamento de processos de negócio (BPM). Em seguida, é apresentada a definição de processos, a classificação destes, além da definição de processos de negócio. Os principais elementos e o ciclo de desenvolvimento do BPM também serão abordados neste capítulo. Estes conceitos são importantes para o entendimento da estrutura e funcionamento do BPM em uma organização. Além dos conceitos relacionados ao BPM, são também apresentadas as principais características dos PAISs, sistemas de informação que permitem suportar os processos de negócio sistematizados e, portanto, aumentar a capacidade de atuação do BPM; visto que a partir da compreensão das características destes sistemas é possível identificar quais ferramentas e quais recursos estão disponíveis às organizações para o gerenciamento de seus processos de negócio. Por fim, serão introduzidos os principais conceitos da teoria de controle de supervisor. Estes conceitos são essenciais para o entendimento e elaboração da arquitetura de controle e supervisão proposta.

2.1 GERENCIAMENTO DE PROCESSOS DE NEGÓCIO - BPM

A constante evolução das tecnologias e sistemas de informação, ao longo do tempo, possibilitou uma aproximação entre as áreas de administração e tecnologia da informação. A relação entre processos e tecnologia da informação tem sido cada vez mais explorada e sua importância tem crescido significativamente. Basicamente, a tecnologia da informação dá suporte à coordenação das atividades dos processos, mas seu papel tem sido continuamente ampliado para apoiar o projeto ou desenho de processos, criar e registrar o conhecimento sobre os processos e para a própria gestão dos processos no dia-a-dia (PAIM et al. 2009).

Neste contexto, surge o BPM. O BPM está fortemente associado à adoção da tecnologia da informação ao modelo organizacional de gestão por processos.

OBPM permite unir a Gestão por Processos e a Tecnologia da Informação a fim de reunir as melhores práticas e ferramentas para definição, análise e melhoria dos processos de negócio. A habilidade para mudar o processo passa a ser mais relevante do que a habilidade para criá-lo, pois ela gera as condições para que toda a cadeia de valor possa ser monitorada, continuamente melhorada e otimizada (SMITH e FINGAR 2003).

Aalst et al. (2003) definem BPM como uma ferramenta para apoiar os processos de negócio utilizando métodos, técnicas e software para projetar, aprovar, controlar e analisar processos operacionais envolvendo humanos, organizações, aplicações, documentos e outras fontes de informação. De Sordi (2008) descreve que o BPM possibilita a capacidade de gerenciamento de uma instância ou ocorrência do processo de negócio ao longo de toda a sua cadeia de processos. Estas definições sugerem o BPM como uma proposta orientada para a gestão de todo o ciclo de vida de processos de negócio com base nas tecnologias de informação.

O foco do BPM é nos sistemas de informação para o gerenciamento de processos que induzem a realização da melhoria dos processos no dia-a-dia das organizações, utilizando como apoio softwares de modelagem de processos, ferramentas CASE – Computer-Aided Software Engineering, plataformas de fluxo de trabalho, SOA – Service Oriented Object, entre outros. O BPM tem sido visto atualmente como uma forma eficaz de promover a integração, a dinâmica, a flexibilidade e a inovação nas organizações, proporcionando desta forma, vantagem competitiva. Os resultados e benefícios atualmente comprovados são (PAIM et al. 2009):

- a) uniformização de entendimentos sobre a forma de trabalho através do uso dos modelos de processo;
- b) melhoria do fluxo de informações a partir da sua identificação nos modelos do processo;
- c) padronização dos processos em função da definição de um referencial de conformidade;
- d) melhoria da gestão organizacional a partir do melhor conhecimento dos processos por meio de indicadores de desempenho, projeto organizacional, sistemas de informação, competências, entre outros;

- e) aumento da compreensão teórica e prática sobre os processos, ampliando a possibilidade de aprimoramento dos mesmos;
- f) redução de tempo e custos dos processos, com enfoque econômico-financeiro;
- g) redução no tempo de atravessamento de produtos;
- h) aumento da satisfação dos clientes;
- i) aumento da produtividade dos trabalhadores;
- j) redução no tempo de identificação dos problemas e a implementação das soluções necessárias.

Além de permitir a análise e melhoria dos processos de negócio, o BPM possibilita também uma análise nos demais tipos de processos que suportam os processos de negócio, como os processos organizacionais / operacionais e os processos gerenciais.

“A proposta do BPM é a da integração e conexão dos sistemas de informação já existentes à camada de gestão por processos de negócio, seja para capturar dados do negócio, para averiguar seu status de operação, para obter dados de seu desempenho, como tempo e rendimento, ou mesmo para acioná-la ou mandar-lhe uma carga de trabalho” (DE SORDI 2008).

No entanto, para que estes resultados e benefícios sejam alcançados pelas organizações, é imprescindível que as ações de análise e modelagem de processos estejam bem fundamentadas e estruturadas, a fim de garantir o rápido diagnóstico acerca da situação dos processos, a elaboração e implantação de melhorias para as possíveis ocorrências e a aplicação eficaz da tecnologia de informação no apoio ao desenvolvimento destas ações.

2.1.1 Definição de processos

Processos segundo Humphrey (2003) é um conjunto definido de passos para a realização de uma tarefa. De acordo com Gonçalves (2000) é qualquer atividade ou conjunto de atividades que recebe uma entrada, adiciona valor a ela, e fornece

uma saída a um cliente específico. Segundo Oliveira (2008) processo é um conjunto de ações ordenadas e integradas para um fim produtivo específico, ao final do qual serão gerados produtos e/ou serviços e/ou informações. Ainda segundo Oliveira (2008) processos são resultados dos sistemas em ação, representados por fluxos de atividades ou eventos.

De acordo com Cruz (2006), processo é um “conjunto de atividades que tem por objetivo transformar insumos (entradas), adicionando-lhes valor por meio de procedimentos, em bens ou serviços (saídas), que serão entregues e devem atender às necessidades dos clientes”. A ABNT (2005), por sua vez, descreve que “quaisquer atividades que recebem entradas e as transformam em saídas podem ser consideradas como um processo”. As definições destes autores convergem para uma idéia única que pode ser representada na figura 2.1.

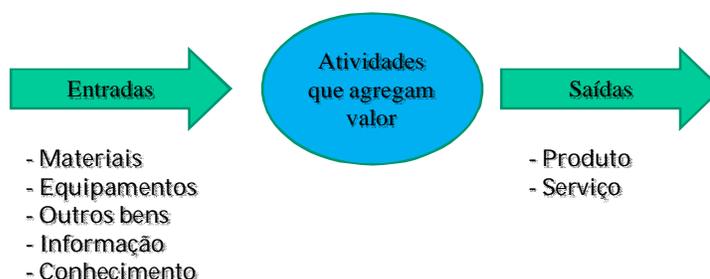


Figura 2.1 - Visão Macro de um Processo

A figura 2.1 ilustra a visão macro de um processo. Nesta visão destacam-se dois dos principais elementos de um processo: as entradas e as saídas. Todo processo, obrigatoriamente deverá ter uma entrada, que será utilizada pelas atividades do processo para gerar uma saída. Desta forma, a saída do processo também é um elemento obrigatório. Além destes dois elementos, outros elementos fazem parte da estrutura de um processo. Estes elementos são representados na tabela 2.1.

Tabela 2.1–Principais elementos dos Processos

Elemento	Definição
Entradas	Tudo o que é introduzido no processo e tem como objetivo ser transformado ou fornecer dados e informações. Pode ser do tipo lógico (dados dos sistemas de informação) ou físico (materiais, bens).
Saídas	Produto e/ou serviço resultante da execução do processo. Pode ser do tipo lógico (dados dos sistemas de informação) ou físico (materiais, bens).
Cientes	São categorizados em Externos e Internos: Clientes externos = são a razão da existência da organização. Os clientes

	externos sempre utilizam o produto / serviço prestado pela organização. Clientes internos = todos os participantes de todos os processos de negócio da organização.
Fornecedores	São categorizados em Externos e Internos: Fornecedores externos = fornecem entradas e estão fora da organização. Fornecedores internos = fornecem entradas e estão dentro da organização.
Atividades	Tem como objetivo processar as entradas para produzir parte do produto do processo.
Papéis Funcionais	Toda atividade tem uma responsabilidade dentro do processo e é esta responsabilidade que o papel funcional deve assumir.

Fonte: Adaptado de Oliveira (2008) e Cruz (2006)

As principais características que podem ser observadas nos processos são descritas a seguir (GONÇALVES 2000; CRUZ 2006; OLIVEIRA 2008; DE SORDI 2008):

- a) utiliza os recursos da organização para atingir os resultados;
- b) pode ocorrer em qualquer tempo, bastando que um gatilho o dispare;
- c) pode produzir milhares de produtos, serviços ou resultados similares;
- d) caracteriza-se pela repetibilidade de ocorrência, podendo coexistir várias instâncias de processo simultaneamente.

Estas características demonstram que os processos estão intrinsecamente relacionados aos fluxos de trabalho existentes nas organizações. Estes fluxos de trabalho quando estruturados na forma de processos permitem um maior controle e ações de melhoria constantes, por meio do aprendizado sobre seu funcionamento.

Para melhorar o entendimento acerca da função/objetivo dos processos dentro de uma organização, é interessante separar os processos de produção dos bens e serviços oferecidos dos demais processos que ocorrem na empresa: os processos relacionados com a gestão da empresa e os de apoio aos processos produtivos (HARRINGTON et al. 1997). Desta forma, a fim de categorizar os tipos de processos existentes em uma organização, Gonçalves (2000) e Cruz (2006) classificam os processos conforme a tabela 2.2.

Tabela 2.2 - Tipos de Processos

Tipos de Processos	Descrição
Negócio / Cliente	Produto e/ou serviço recebido pelo cliente externo
Organizacionais/Integração Organizacional	Viabilizam o funcionamento coordenado dos vários subsistemas da organização e garantem suporte aos processos de negócio
Gerenciais	Medição e ajuste do desempenho da organização

Fonte: Adaptado de Gonçalves (2000) e Cruz (2006)

Esta classificação pode ser utilizada principalmente nas atividades de identificação e mapeamento dos processos, pois pode ser utilizada como um critério de priorização para a melhoria dos processos. “Melhorar processos é uma ação básica para as organizações responderem às mudanças que ocorrem constantemente em seu ambiente de atuação e para manter o sistema produtivo competitivo” (PAIM et al. 2009). “Uma empresa só funciona de maneira eficaz quando é capaz de identificar e gerir suas numerosas atividades interligadas” (OLIVEIRA 2008).

A abordagem utilizada pela presente pesquisa abrange os processos de negócio que são fortemente suportados por sistemas de informação. De acordo com Oliveira (2008) os processos de negócio são atividades coordenadas que envolvem: pessoas, procedimentos, recursos e tecnologia. Um processo de negócio consiste em cinco elementos (Lin et al. 2002):

- a) clientes;
- b) atividades;
- c) estas atividades são voltadas para criar valor para seus clientes;
- d) as atividades são operadas por atores que podem ser seres humanos ou máquinas;
- e) freqüentemente envolve várias unidades organizacionais que são responsáveis por todo o processo.

Segundo Weske (2007) um processo de negócio consiste em um conjunto de atividades que são executadas em coordenação em um ambiente organizacional e técnico. O ambiente técnico mencionado refere-se ao suporte ao processo por meio de tecnologias de informação, mais especificamente de um sistema de informação, um PAIS. Weske (2007) enfatiza ainda que os processos de negócio são essenciais para a compreensão de como as empresas operam. Segundo ele, os processos de negócio também desempenham um papel importante na concepção e realização de sistemas de informação flexível. Estes sistemas de informação constituem a base técnica para a criação e adaptação de funcionalidades a fim de atender às novas exigências do mercado.

2.1.2 Ciclo de vida e elementos do BPM

As principais características acerca do ciclo de vida e elementos do BPM apresentados neste tópico tem como base os trabalhos de Kirchmer (2006), Smith e Fingar (2003), Jost e Scheer (2002), Muehlen e Ho (2005), Aalst et al.(2003), Weske (2007), Magalhães et al. (2004), Oliveira (2008), De Sordi (2008) e Paim et al. (2009). De acordo com estes autores, o ciclo de vida de BPM tipicamente envolve as seguintes etapas: projeto e análise, implementação, execução e monitoramento; sendo que estas etapas são realizadas de forma contínua, conforme ilustrado na figura 2.2.



Figura 2.2 - Ciclo de vida do BPM

Cada etapa do ciclo de vida do BPM dispõe de elementos que irão possibilitar o entendimento do cenário em que se encontram os processos de negócio e permitir a análise, implementação, execução e monitoramento da execução do processo a fim de avaliar se a aplicação das regras de negócio pré-definidas e, necessárias para o alcance do melhor resultado do processo, estão sendo devidamente atendidas. Além disso, com base na etapa de monitoramento, será possível repensar e reavaliar as estratégias adotadas anteriormente e praticar a melhoria contínua nos processos, reiniciando o ciclo de vida de cada processo por meio da etapa de projeto e análise. Os principais elementos e ferramentas disponíveis em cada etapa do ciclo de vida são ilustrados na figura 2.3 e discutidos nos parágrafos seguintes.

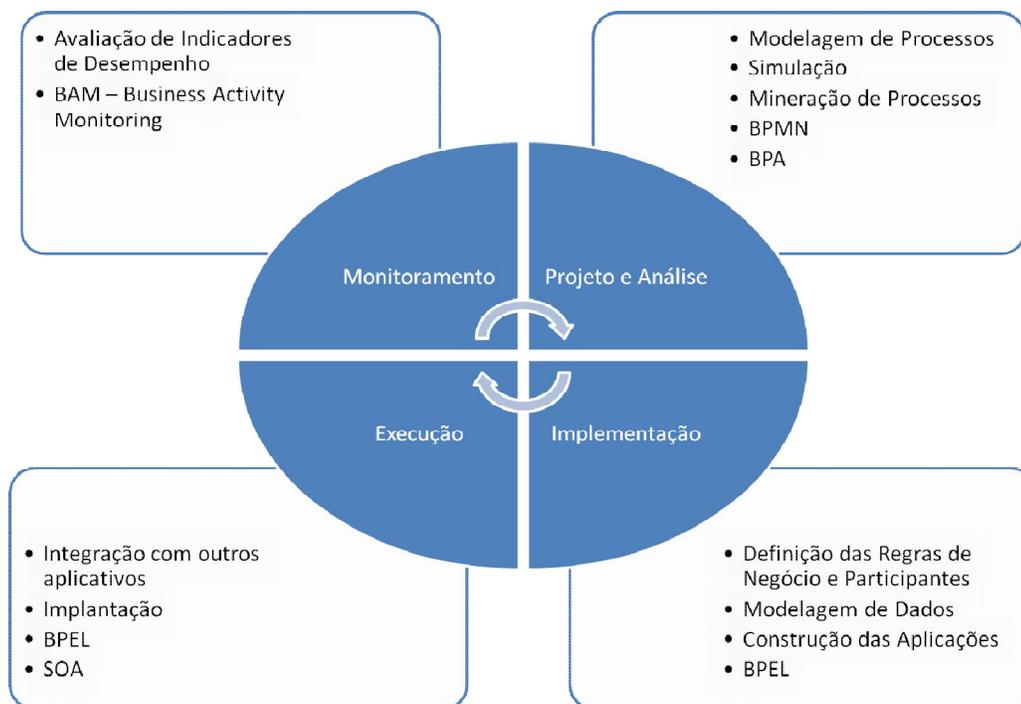


Figura 2.3 - Ciclo de vida e principais elementos e ferramentas do BPM

Segundo Weske (2007), o ciclo de vida do BPM tem início na etapa de projeto e análise, em que um levantamento acerca do ambiente organizacional e técnico onde se encontra o processo de negócio é realizado. Com base nesse levantamento, processos de negócio são identificados, analisados, validados e representados por modelos de processos de negócio. Além disso, na etapa de projeto/análise são definidos os objetivos que se pretendem alcançar, bem como a análise dos pontos de falhas nos processos que causam prejuízos à organização. Os principais elementos e ferramentas que fazem parte desta etapa são:

- a) Modelagem de processos;
- b) Simulação;
- c) Mineração de processos;
- d) BPMN – Business Process Modeling Notation;
- e) BPA – Business Process Analysis.

De acordo com Oliveira (2008), o objetivo da modelagem de processos é a análise e a documentação da funcionalidade dos processos de negócio de forma a avaliar se tais processos cumprem com seus objetivos, identificar os fatores críticos de sucesso, analisar seu comportamento e desempenho, destacando os pontos

fracos, futuros objetos de ações de melhoria. Além disso, a construção de um modelo permite a avaliação dos processos de negócio envolvidos. Isso viabiliza a formulação de mudanças, estruturadas de forma a garantir o cumprimento da missão organizacional e o atendimento das estratégias empresariais necessárias ao sucesso em seu ramo de atuação (OLIVEIRA 2008). Um modelo é a representação da realidade por meio da construção de diagramas, protótipos, ou qualquer recurso que permita visualizar o funcionamento do objeto a ser estudado, neste caso, o processo de negócio a ser analisado; permitindo a compreensão de suas características e a simulação do comportamento submetido à ação de eventos programados (OLIVEIRA 2008). Para Magalhães et al. (2004), o modelo de negócio é um conjunto de visões que representam características diferentes de um ou mais aspectos específicos do negócio. Um modelo de negócio deve refletir o processo da organização. Ao utilizar um modelo, a organização poderá analisar com detalhes a estrutura representada e definir cenários que possam contribuir para a evolução e melhoria do processo analisado.

Para tanto, ao elaborar um modelo de processo deve-se verificar a abrangência de diferentes perspectivas. De acordo com Aalst (2003) quatro perspectivas são consideradas essenciais para a modelagem: perspectiva do processo, da informação, dos recursos e das atividades. A perspectiva de processo descreve o fluxo do trabalho, ou seja, a ordenação de tarefas. A perspectiva de informação descreve os dados que são utilizados pelo processo. A de recurso descreve a estrutura da organização e identifica recursos, papéis, e de grupos. E, por fim, a de tarefa descreve o conteúdo de cada uma das etapas do processo. A figura 2.4 ilustra a relevância destas perspectivas e a forma de integração entre elas.



Figura 2.4– Perspectivas essenciais para a modelagem de Processos de Negócio

Fonte: AALST (2002) adaptado pelo autor

Para iniciar a modelagem de um processo, é essencial saber quais informações são relevantes para a sua compreensão como um todo. A identificação destas informações também será relevante para a escolha da notação a ser utilizada para o mapeamento do processo. Em grande parte das notações existentes, estas informações podem ser encontradas categorizadas pelos seguintes itens: atividade, comportamento, recurso, relação entre atividade, agente, informação, entidade de informação, evento, validação e procedimento de modelagem.

Após a elaboração e/ou revisão do modelo do processo, caso este já exista, é possível utilizar a simulação para avaliar e validar o modelo do processo elaborado. Segundo Oliveira (2008), a simulação de processo permite a criação de cenários com o intuito de mensurar os impactos gerados pela variação de aspectos comportamentais inerentes ao funcionamento do processo. A interpretação dos resultados apresentados pela simulação do processo poderá sugerir a adoção de medidas e/ou ações para a melhoria do processo. De Sordi (2008) descreve ainda que a simulação dinâmica de processos permite a realização de ajustes até que se obtenha um resultado desejado. A cada nova simulação, novos valores para cada um dos indicadores de desempenho são atribuídos e estes valores permitem a realização de comparações entre os diversos resultados obtidos nas diversas simulações. As técnicas de simulação podem ser usadas para apoiar a validação do processo, visto que um fluxo de execução indesejado pode ser simulado e o seu resultado avaliado por meio de dados estatísticos. A simulação de processos de negócio também permite verificar se o processo realmente atende as situações e os cenários levantados durante a análise e modelagem do processo (WESKE 2007).

Ainda na etapa de projeto e análise, é possível utilizar as técnicas de mineração de processos para avaliar os registros históricos das execuções dos processos. As técnicas de mineração de processos permitem extrair informações de processos de negócio por meio de logs de eventos (HARRINGTON et al. 1997). Informações como atividades do processo, responsáveis pela execução das atividades, data em que as atividades foram realizadas e eventos que desencadearam uma atividade, são alguns exemplos de informações que podem ser extraídas por meio da técnica. O objetivo da mineração é, através da coleta de dados em tempo de execução, inverter o processo e apoiar a concepção e análise do fluxo de trabalho utilizado. A idéia básica da mineração de processos é

diagnosticar o conhecimento do processo de negócio por meio dos logs de eventos (AASLT e MEDEIROS 2004). Em Aalst, Weijters e Maruster (2004), afirma-se ainda que a mineração de processos pode ser utilizada em duas situações: (1) como uma ferramenta para descobrir como as pessoas e / ou procedimentos realmente funcionam e, (2) para análise Delta, comparando o processo real com o seu modelo.

A etapa de projeto e análise pode ser realizada com o suporte de ferramentas como BPMN – Business Process Modeling Notation e BPA – Business Process Analysis. O BPMN fornece uma notação gráfica para a expressão dos processos de negócios em um diagrama – Business Process Diagram (BPD). O principal objetivo do BPMN é permitir que o gerenciamento de processos de negócio seja uma tarefa intuitiva e que possa ser utilizada tanto por usuários técnicos, quanto por usuários de negócios (BPML, 2004), para a elaboração de modelos de processos de negócio. Além do BPMN, existem também outras notações que podem ser utilizadas para a modelagem de processos (Petri nets, EPCs, Workflow nets, YAWL, BPMN, etc.). Na presente pesquisa, além do BPMN também será utilizado o YAWL – Yet Another Workflow Language. O BPA, por sua vez, é uma ferramenta que permitirá realizar as técnicas de análise, verificação e simulação mencionadas e ainda auxiliar na elaboração do modelo de processo ideal para o alcance dos objetivos das organizações.

A etapa de implementação tem como objetivo implementar a proposta elaborada. Na etapa de implementação é colocado em prática o modelo do processo elaborado na etapa de projeto e análise. Nesta etapa todos os recursos necessários são definidos, além das regras de negócio como novas normas e procedimentos necessários para a execução dos processos. Nesta etapa também são configurados os meios requeridos para a implantação do processo e, posteriormente à estas definições e configurações tem início a utilização do novo processo. Os principais elementos e ferramentas que fazem parte desta etapa são:

- a) Definição das Regras de Negócio e Participantes;
- b) Modelagem de Dados;
- c) Construção das Aplicações;
- d) BRE – Business Rules Engine;
- e) BPEL – Business Process Execution Language.

Na fase de definição das regras de negócio e participantes são realizadas as preparações dos recursos: quais funções podem ser distribuídas dentro do processo, a localização dos recursos (alocação) dentro da estrutura e associações dos recursos; além da definição de políticas e práticas que irão garantir a execução do fluxo do processo a fim de alcançar o resultado planejado. Aalst e Hee (2002) descrevem que em cada trabalho, são alocados recursos que podem ser representados por máquinas e/ou pessoas e que nesta linha de análise são muito importantes para a definição de eficiência e efetividade do fluxo do processo.

A determinação dos recursos dentro de um processo, para Aalst e Hee (2002) está representado por dois tipos diferentes de classificação: (1) baseado sobre as propriedades funcionais e (2) sobre a posição dentro da organização. As propriedades funcionais também conhecidas como unidades organizacionais representam a localização dentro da organização “onde” as atividades são realizadas; são considerados como os departamentos, os processos ou as unidades gerenciais, onde estão alocadas as pessoas (recursos) do processo. Estes recursos quando alocados dentro dos devidos processos, possuem papéis, determinações, ou características que podem ser definidas como hierarquias do processo, que representam as posições destes recursos dentro das unidades organizacionais (AALST e HEE 2002). A figura 2.5 apresenta uma visão da classificação de recursos envolvendo identificação de unidades organizacionais, departamentos e pessoas (recursos).

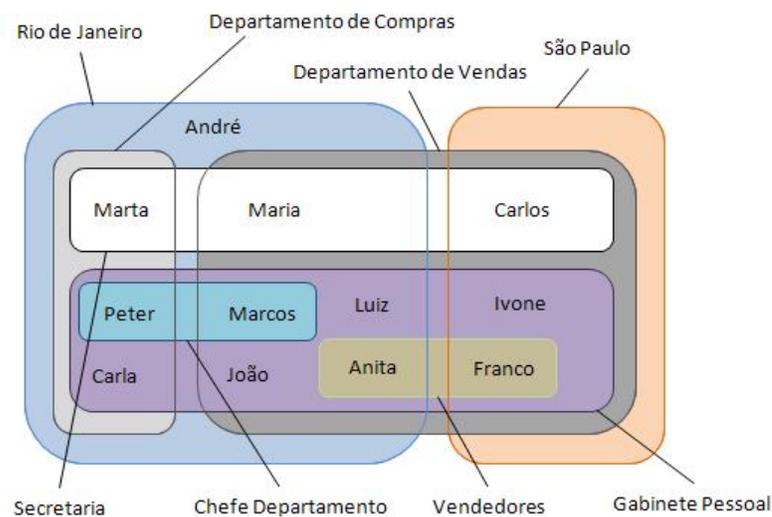


Figura 2.5 - Classificação de recursos

Fonte: AALST e HEE (2002) adaptado pelo autor

A implementação de uma regra de negócio se dá por meio de documentos utilizados na operação ou pela tecnologia, mais especificamente por meio dos algoritmos de softwares utilizados pelo processo (DE SORDI 2008). Implementadas via tecnologia ou documentos, as regras de negócio representam limitações desejáveis no comportamento humano ou nos sistemas de informação (softwares), permitindo que a organização atenda aos requisitos da legislação, de órgãos ou entidades reguladoras, as políticas da empresa e aos demais procedimentos que a empresa tenha de seguir (DE SORDI 2008). Porém para que as regras de negócio possam ser devidamente implementadas, é fundamental o entendimento das abstrações de eventos, estados, entidades e atributos definidos para o fluxo do processo. Portanto, faz-se necessário realizar a análise e modelagem dos dados que serão utilizados pelo processo. Os diversos dados manipulados durante a operação de um processo devem ser bem descritos e documentados (DE SORDI 2008). A modelagem de dados possibilita esta documentação dos dados manipulados pelo processo e permite a compreensão de como os dados são utilizados e como são processados durante a execução do fluxo do processo.

A etapa de implementação do processo pode ser realizada com o suporte de ferramentas como BRE – Business Rules Engine e BPEL – Business Process Execution Language. O BRE é uma ferramenta que permitirá realizar a definição das regras de negócio com base no modelo de processo elaborado. O BPEL, por sua vez, é uma ferramenta que permite implementar (automatizar) o modelo do processo de negócio e disponibilizá-lo para execução. Por meio do BPEL também é possível implementar as integrações entre outros sistemas. Além do BPEL, existem também outras iniciativas para a automatização e integração dos processos.

A etapa de execução permite a implantação e a execução do fluxo do processo, de acordo com a nova proposta implementada. Os principais elementos e ferramentas que fazem parte desta etapa são:

- a) Integração com outros aplicativos;
- b) Implantação;
- c) BPEL – Business Process Execution Language;
- d) SOA –Service Oriented Architecture.

Durante a etapa de execução do processo, instâncias do processo de negócio são criadas a cada necessidade de execução do processo a fim de gerar ou satisfazer alguma necessidade de negócio da empresa. Em grande parte dos processos, para que a execução das instâncias dos processos seja realizada, muitas vezes é necessário o compartilhamento e a busca de dados e informações de outros sistemas disponíveis na empresa. Para isso, utiliza-se então, a integração do processo implementado com os sistemas de informação existentes na empresa. O SOA é uma das propostas existentes que permite a integração dos dados gerados pelos diversos processos das empresas e que estão armazenados em ambientes tecnológicos diversos. Os sistemas de gestão de processos de negócio – BPMS – controlam a execução das instâncias dos processos de negócio, tal como definido no modelo do processo. A execução do processo deve atender a uma correta orquestração de processos, garantindo, desta forma, que as atividades do processo sejam executadas de acordo com as limitações de execução especificadas no modelo de processo (WESKE 2007).

Por fim, na etapa de monitoramento, a execução do fluxo do processo será monitorada com o intuito de garantir o controle e a geração de informações que posteriormente realimentarão o ciclo de vida do BPM. O monitoramento do processo é um importante mecanismo para fornecer informações precisas sobre o status das instâncias dos processos de negócio (WESKE 2007). Acompanhar ou monitorar a execução dos processos tem como objetivo identificar a trajetória de desempenho e tornar possível tanto a decisão de controlar o processo quanto a formação de um histórico de desempenho que permita melhor conhecer o processo (PAIM et al. 2009). Os principais elementos e ferramentas que fazem parte desta etapa são:

- a) Avaliação de Indicadores de Desempenho;
- b) BAM - Business Activity Monitoring.

A avaliação de desempenho tem como objetivo analisar se um processo pode ter seus limites de desempenho reduzidos, mantidos ou mesmo ampliados (PAIM et al. 2009). O registro do desempenho, a comparação com referenciais internos e externos e a identificação das causas de impactos significativos sobre os processos criam, em conjunto, uma base para julgar a trajetória de desempenho e, assim, avaliar os processos (PAIM et al. 2009). Um dos desafios das organizações orientadas a processos é ter indicadores de desempenho eficientes que tornem as

organizações aptas a perceber com rapidez as condições que estão começando a afetar seus processos de negócio e a organização. Dessa maneira, as organizações podem realizar o quanto antes os ajustes necessários em seus processos de negócio com o propósito de obter os resultados esperados (DE SORDI 2008).

Para auxiliar na etapa de monitoramento, é possível utilizar ferramentas como BAM – Business Activity Monitoring. Este tipo de ferramenta permitirá realizar o acompanhamento e o monitoramento da execução das atividades dos processos.

2.1.3 Sistemas de gestão de processos – BPMS

De acordo com Paim et al. (2009), a consolidação e a evolução do conceito de gestão de processos naturalmente resultaram no desenvolvimento de tecnologias de suporte à gestão de processos, como por exemplo, o fluxo de trabalho, trabalho colaborativo e as aplicações de integração da empresa (Enterprise Application Integration – EAI). Segundo estes autores, conforme o conceito de gestão de processos foi sendo ampliado e consolidado, surgiram mais ferramentas usadas para desenhar modelos de processos de negócio, processar o fluxo de dados, regras de negócio, otimizar, monitorar e manter vários processos que ocorrem dentro de uma organização. Estas ferramentas que surgiram e continuam a surgir são conhecidas como sistemas de gestão de processos – BPMS.

Paim et al. (2009) descreve que os BPMS têm sido utilizados com o objetivo maior de atingir a conformidade aos processos. Os BPMS, segundo Smith e Fingar (2003), possibilitam que as organizações modelem, disponibilizem e gerenciem processos críticos para a sua missão, que podem estar distribuídos entre múltiplos aplicativos da empresa, departamentos corporativos e parceiros de negócio.

Os BPMS podem ser entendidos como um conjunto de instrumentos que buscam a melhoria do sistema de gestão, contribuindo para a implementação de mudanças que tornem ou mantenham a empresa competitiva com os fluxos de trabalho claramente definidos, automatizados e racionais (VERNER 2004).

As ferramentas mencionadas no tópico 2.1.2 geralmente fazem parte das soluções de BPMS disponíveis atualmente. Estas ferramentas têm funções de análise e de representação que concretizam a verificação, a avaliação e a

modificação dos processos e de suas estruturas organizacionais (PAIM et al. 2009). Os BPMS interligam pessoas e processos, gerenciam a transformação e o acesso à informação, tratam exceções e orquestram o fluxo de processos (PAIM et al. 2009).

Qualquer BPMS, instalado e customizado, deve ser capaz de (OLIVEIRA 2008):

- a) captar e identificar – modelar – os processos críticos e necessários à gestão do negócio;
- b) entender, aceitar e operar o esquema de identificação, o seqüenciamento e a interação destes processos;
- c) tornar possível a integração do sistema de gestão de processos com o ambiente de TI;
- d) aceitar o conjunto de critérios e métodos (metodologia) adotados pela organização, visando assegurar a efetiva operação e o monitoramento desses processos;
- e) fornecer e disponibilizar, a tempo e na hora certa, informações sobre esses processos;
- f) possibilitar o monitoramento das atividades – BAM –, monitorar o funcionamento e desempenho dos processos;
- g) fornecer ferramentas para análise da estrutura atual, simulação e otimização de processos;
- h) fornecer recursos e facilidades para a implementação de ações, visando à obtenção de resultados planejados e a melhoria contínua desses processos.

2.1.4 Arquitetura Orientada a Serviço – SOA

Com o avanço das tecnologias e a grande disseminação da informação dentro das empresas, muitas aplicações de softwares foram, e ainda são desenvolvidas com o objetivo principal de automatizar e/ou operacionalizar tarefas repetitivas. Neste processo, porém, grande parte dos dados e funções destas aplicações são duplicados, tornando a manutenção destes uma tarefa complexa e com o custo consideravelmente elevado.

Para minimizar a complexidade e custos das tarefas de manutenção e propor uma arquitetura enxuta na infra-estrutura da empresa, o conceito de SOA pode ser implementado. SOA é um conceito que propõe a organização das aplicações de software, a fim de representar os Processos, as Atividades ou as Tarefas de Negócio da empresa de forma direta. Por meio de SOA são criados “serviços” que irão permitir a integração e redução da redundância de dados e aplicações, garantindo um alinhamento estratégico da empresa com os recursos e benefícios das tecnologias da informação. Um serviço é um componente de software que pode ser acessado por meio de uma rede para fornecer o recurso a um componente de software requisitante (SRINIVASAN e TREADWELL 2005).

De acordo com Booth et al. (2004), SOA é um tipo de arquitetura baseado em sistemas distribuídos. Um sistema distribuído consiste de diversos componentes discretos de software que devem trabalhar em conjunto para executar algumas tarefas. Além disso, os componentes em um sistema distribuído não operam necessariamente no mesmo ambiente de transformação, por isso, devem se comunicar por meio de uma rede, utilizando hardware, software e protocolos de rede. Srinivasan e Treadwell (2005) citam como principais características da arquitetura SOA:

- a) os serviços podem ser utilizados individualmente ou podem ser integrados com outros serviços, isto promove a reutilização de funcionalidade entre os serviços;
- b) a comunicação entre os serviços e seus clientes (componentes de software que requisitam o serviço) é feita por troca de mensagens, o que permite interoperabilidade entre as diversas tecnologias utilizadas no ambiente do serviço e das aplicações clientes;
- c) os serviços podem participar de um fluxo de trabalho, onde a ordem na qual as mensagens são enviadas e recebidas afeta o resultado das operações realizadas por um serviço. Este conceito é definido como “serviço de coreografia”;
- d) os serviços podem ser auto-suficientes ou podem depender da disponibilidade de outros serviços ou recursos, como um banco de dados. No caso mais simples, um serviço pode apenas desempenhar um cálculo, por exemplo;

e) os serviços publicam as informações em relação as suas capacidades, interfaces, regras e protocolos utilizados. Isto permite às aplicações clientes utilizar da melhor forma os serviços disponíveis em um ambiente.

Atualmente a utilização de *Web Services* para o desenvolvimento da arquitetura SOA tornou-se bastante eficaz. De acordo com Booth et al. (2004) *Web Services* é um sistema de software responsável por proporcionar a interação entre duas máquinas através de uma rede. Para possibilitar essa interação uma interface descrita em um formato específico – Web Services Description Language (WSDL) – permite que sistemas interajam com um *Web Service* usando essa interface e enviando mensagens SOAP (Simple Object Access Protocol) ou utilizando outros protocolos. As mensagens SOAP basicamente são documentos XML serializados seguindo o padrão W3C enviados por meio de um protocolo de rede.

Na presente pesquisa o conceito de SOA e a tecnologia *Web Services* foi utilizada para possibilitar a comunicação entre os PAISs e o SCS. Por meio do *Web Services* foi possível enviar os logs de eventos da execução atual do processo dos PAISs para o SCS e enviar o guia para execução das atividades do SCS para os PAISs.

2.2 PROCESS-AWARE INFORMATION SYSTEMS – PAIS

Atualmente é difícil imaginar uma organização de médio ou grande porte sem um eficiente software de gestão de operações (CLETO 2002). As decisões acerca do uso da Tecnologia da Informação (TI) para apoiar as atividades das empresas tornaram-se progressivamente mais importantes, pois podem ter grande impacto estratégico, em especial no cenário de competição globalizada, na qual a operação just-in-time passa a ser fator de obtenção e de manutenção de vantagens competitivas (LAURINDO et al. 2002).

A devida aplicação de sistemas de informação e a integração destes sistemas pode ser um fator determinante para a melhoria de potencial e eficiência de uma empresa. De acordo com Cleto (2002), a integração cada vez maior dos vários elos da cadeia produtiva (fornecedores e clientes) possibilita um desempenho mais

eficiente e competitivo do setor como um todo; requisito importante para o seu fortalecimento em nível nacional e o seu sucesso no mercado internacional.

Neste cenário, encontram-se os sistemas de informação baseados em processos, ou Process-Aware Information System (PAIS). O PAIS é um sistema de software que gerencia e executa processos operacionais envolvendo pessoas, aplicações e / ou informação fontes, com base em modelos de processo (DUMAS et al. 2005). Exemplos clássicos de PAISs são os Workflow Management Systems (WFMS) e sistemas de Business Process Management (BPMS) (AALST 2009a). Porém, diferente dos WFMS que tem como objetivo apenas a execução do modelo do processo, os BPMS, como já mencionado nos tópicos anteriores, tendem a dar mais apoio às diversas formas de análise e de gestão, como por exemplo, a simulação e o acompanhamento dos seus processos.

Os PAISs desempenham um papel importante no ciclo de vida do BPM. Com a utilização dos PAISs é possível aliar o conhecimento da tecnologia da informação com o conhecimento de gestão e automatizar os processos de negócio. A relação entre o PAIS, o BPM e o WFM é ilustrada na figura 2.6.

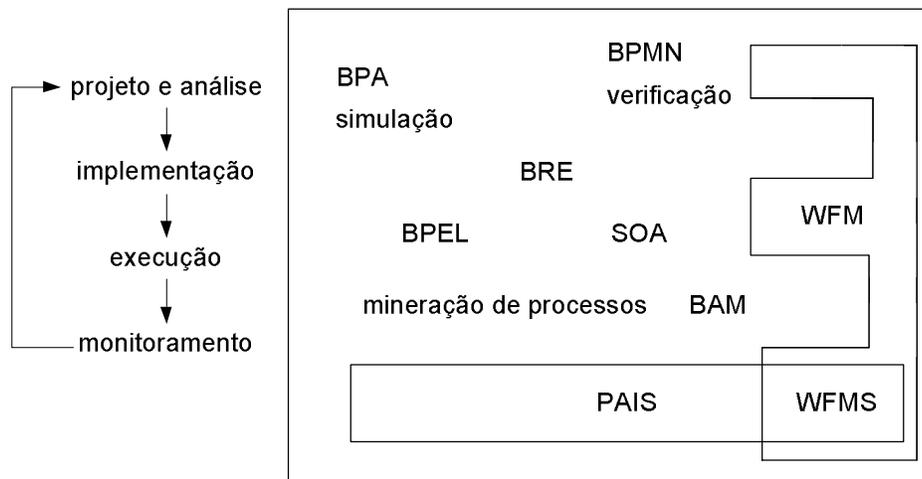


Figura 2.6–Relação PAIS, BPM e WFM

Fonte: AALST (2009b) adaptado pelo autor

A figura 2.6 mostra a relação entre PAIS, BPM, e WFM. Neste contexto, o termo PAIS se refere ao software, enquanto os termos de BPM e WFM se referem aos campos de conhecimento em que PAISs podem ser usados. Os Workflow Management Systems (WFMSs) podem ser vistos como um tipo particular de PAIS

onde a ênfase está na automatização de processos, em vez de projeto e análise. De acordo com Aalst (2009b) a definição de WFMS poderia ser: "um sistema de software genérico, que é impulsionado pelo desenho, criação e gerenciamento de processos de negócio operacionais". Com base nesta definição entende-se que um WFMS é um tipo de PAIS genérico cujo objetivo único é gerenciar as instâncias do processo e garantir a execução do fluxo de trabalho de acordo com o modelo do processo projetado. Portanto, diferente dos BPMS, os WFMS geralmente não possuem ferramentas de análise como BPA, e de monitoramento como BAM, sendo sua principal função a execução do fluxo de trabalho do processo. A figura 2.7 ilustra a arquitetura dos WFMS.

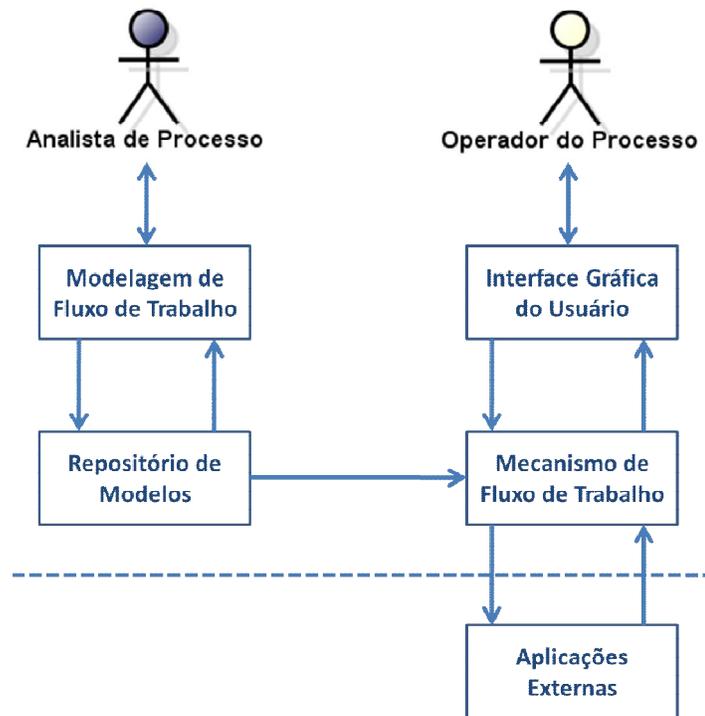


Figura 2.7–Arquitetura WFMS

Fonte: WESKE (2007) adaptado pelo autor

A arquitetura do WFMS contém os seguintes subsistemas e funções, cujas competências estão descritas a seguir (WESKE 2007):

- a) A Modelagem de Fluxo de Trabalho fornece meios para modelar os aspectos técnicos dos processos de negócio implementados ou a serem implementados. Para cada atividade no modelo operacional de processos de negócio realizado pelo software, a especificação detalhada do ambiente de execução do software precisa ser fornecida. A atividade de modelagem

é desempenhada por meio de interação humana, neste caso, pelo papel do Analista de Processo;

- b) Os modelos de fluxo de trabalho são armazenados em um repositório de modelos de fluxo de trabalho. Este repositório contém o conjunto de modelos de fluxo de trabalho da empresa, tornando-se um recurso importante para a gestão dos processos e do conhecimento destes processos na empresa;
- c) O mecanismo de fluxo de trabalho é responsável por permitir a execução e o gerenciamento das atividades do modelo definido para os processos de negócio. Para cada nova necessidade de execução do processo, o mecanismo cria uma instância do processo e permite a execução do fluxo de trabalho por meio dos operadores do processo. As informações sobre os participantes (operadores) do processo, suas habilidades e competências, pode ser usado pelo mecanismo de fluxo de trabalho para habilitar as atividades somente para os operadores que estão disponíveis e possuem o perfil adequado para a realização dessas atividades;
- d) A execução das atividades do fluxo de trabalho é realizada pelos operadores do processo, por meio da interface gráfica do usuário. Todo o funcionamento da interface gráfica do usuário é gerenciado pelo mecanismo de fluxo de trabalho;
- e) Por fim, o mecanismo de fluxo de trabalho pode possuir integração com outras aplicações da empresa. Esta integração é realizada por meio de uma interface de integração utilizando SOA e/ou outros recursos de integração de sistemas heterogêneos. A interface de integração com aplicativos externos fornece as informações técnicas para chamar os aplicativos que vão realizar atividades específicas do fluxo de trabalho do processo, ou ainda fornecer informações que permitam a execução das atividades. Esta interface deve facilitar a invocação de aplicações em plataformas de software heterogêneas.

De acordo com Aaslt (2009a), existem basicamente duas formas para o desenvolvimento de um PAIS: (1) desenvolver um sistema de apoio específico do processo, ou (2) configurar um sistema genérico. No primeiro caso, uma organização constrói seu próprio sistema de apoio do processo "do zero" com o objetivo

específico de apoiar os seus processos. Este sistema específico da organização pode ser tão simples como uma biblioteca de software que fornece rotinas para incorporar o conhecimento do processo em aplicações, ou pode assumir a forma de uma plataforma de execução do processo a fim de fornecer recursos para a definição, análise, implantação, execução e acompanhamento dos processos.

A primeira abordagem é conhecida como “ad hoc” e assegura que o sistema resultante se adapta às necessidades da organização e as especificidades de seus processos. No entanto, o custo de investimento inicial desta abordagem pode ser muito alta para algumas organizações, e o sistema resultante pode não ser escalável. Como sempre são introduzidos novos processos, os processos existentes se tornam mais sofisticados, tornando o desenvolvimento do PAIS cada vez mais específico e complexo para adaptá-lo a processos que necessitam de novas demandas constantemente.

Na segunda abordagem, os sistemas utilizados geralmente não são desenvolvidos pelas organizações, embora haja casos em que um sistema específico da organização tenha posteriormente evoluído para um sistema semelhante a um produto de software genérico. Um exemplo típico de um produto de software genérico é o WFMS. Os WFMSs são genéricos na medida em que não incorporam informações sobre a estrutura e processos de qualquer organização em particular. Para utilizar um WFMS, a organização precisa configurar os processos definindo todas as entidades organizacionais e demais informações para que seja possível a execução do fluxo do processo de acordo com o modelo elaborado.

2.2.1 Tipos de PAIS

Os PAISs podem ser classificados de acordo com a natureza dos participantes e/ou recursos do processo. Esta classificação considera principalmente a questão dos participantes / recursos serem humanos ou aplicações de software. De acordo com Aast (2009a), os PAIS podem ser classificados da seguinte forma:

- a) P2P, pessoa-pessoa;
- b) P2A, pessoa-aplicação;

c) A2A, aplicação-aplicação.

Em P2P os participantes do processo são principalmente as pessoas, ou seja, os processos envolvem principalmente as tarefas que requerem a intervenção humana. Geralmente, os processos suportados por essas ferramentas não envolvem tarefas totalmente automatizadas e são realizadas pelos participantes com apoio destas aplicações. Exemplos deste tipo de ferramenta são: Microsoft Office Project que auxilia no gerenciamento de projetos, Microsoft Outlook que auxilia no gerenciamento de e-mails, entre outros. Os PAISs do tipo P2P são altamente flexíveis, porém, em contrapartida, não possui suporte ao usuário na execução dos processos, exigindo desta forma, um alto nível de conhecimento do usuário para a execução dos processos.

Os PAISs classificados como P2A, são aqueles que envolvem as tarefas humanas e as interações entre as pessoas, além das tarefas e as interações que envolvem aplicações que funcionem sem intervenção humana. De acordo com Aalst (2009a), os WFMSs se inserem nesta categoria, uma vez que visam principalmente fazer as pessoas e os aplicativos funcionarem de forma integrada. Outros exemplos deste tipo de ferramenta são: os BPMS, os sistemas de gestão empresarial como SAP R/3, entre outros. Os PAIS do tipo P2A não são tão flexíveis quanto os do tipo P2P, sendo sua flexibilidade medida com base no nível de suporte que é fornecido ao usuário.

Por fim, os A2A são aqueles PAIS que envolvem apenas tarefas executadas por sistemas de software. Segundo Aalst (2009a) tais ferramentas são típicas da área de computação distribuída e, em particular, a integração de aplicações distribuídas. Sistemas de processamento de transações, as plataformas de EAI e integração de servidores baseados na Web são projetados para suportar os processos A2A. Note-se que às vezes a lógica desses processos é capturada por modelos de processo explícito, e outras vezes é implicitamente codificado em programas que participam do processo. Como os recursos que participam nos processos de A2A são aplicações, e estes podem partilhar bases de dados comuns, um aspecto importante que surge neste tipo de processo é garantir certas propriedades transacionais, tal como definido no âmbito dos sistemas de gerenciamento de banco de dados (SGBD).

No contexto do presente trabalho, o tipo de PAIS utilizado para a aplicação da abordagem proposta é classificado como P2A.

2.2.2 Flexibilidade no PAIS

Inicialmente o foco dos PAISs era sobre a utilização das tecnologias de suporte ao processo, como por exemplo, sistema de fluxo de trabalho, para automatizar processos operacionais envolvendo pessoas e aplicações dentro de uma organização. No entanto, ao automatizar um processo e utilizar um PAIS como suporte, é importante identificar o grau de automação necessário para o processo, a fim de identificar a classificação mais adequada de PAIS para este processo.

Esta identificação é necessária, pois “o dinamismo dos atuais ambientes de negócios gera constantes alterações nas condições do mercado que obrigam os executivos a reagir o mais rápido possível, o que implica alterações nas operações da empresa” (DE SORDI 2008). Em contrapartida, “o ambiente de sistemas de informação intrínseco na operação e nos processos da empresa sempre foi um dos recursos operacionais mais inflexíveis. A lógica implementada pelos softwares, a estrutura de dados implantada pela base de dados e os meios de comunicação de um software para com os demais sistemas são alguns exemplos de componentes de um sistema de informação que apresentam alta complexidade, que demandam muito tempo para sua alteração e, quando alterados, aumentam sensivelmente a exposição a falhas e erros no processo, ou seja, aumento do risco” (DE SORDI 2008). Portanto, é importante que as organizações estejam atentas ao ambiente de processo e características que o envolvem para que o PAIS selecionado como suporte para os processos possa atender a todas as necessidades deste processo.

A figura 2.8 ilustra um comparativo entre os tipos de PAIS existentes nas organizações e as questões que envolvem: Flexibilidade, Desempenho, Suporte e esforço de Projeto. Conforme demonstrado na figura e mencionado no tópico sobre os tipos de PAIS, os PAIS classificados como P2P são aqueles que possuem um alto nível de flexibilidade e desempenho. No entanto, este tipo de ferramenta possui um nível muito baixo (ou quase nenhum) de suporte para os usuários durante a execução das atividades dos processos. Pode-se observar também, por meio da

figura, que conforme o suporte aumenta nas ferramentas, a flexibilidade diminui. A flexibilidade neste caso, refere-se à capacidade de modificação permitida para a estrutura de implementação dos processos, a fim de adequá-los às suas demandas. O suporte, por sua vez, refere-se à capacidade de provimento de recursos para análise e orientação para o usuário. Com base neste cenário, pode-se afirmar então que os PAISs oferecem apoio ou flexibilidade, mas não os dois.

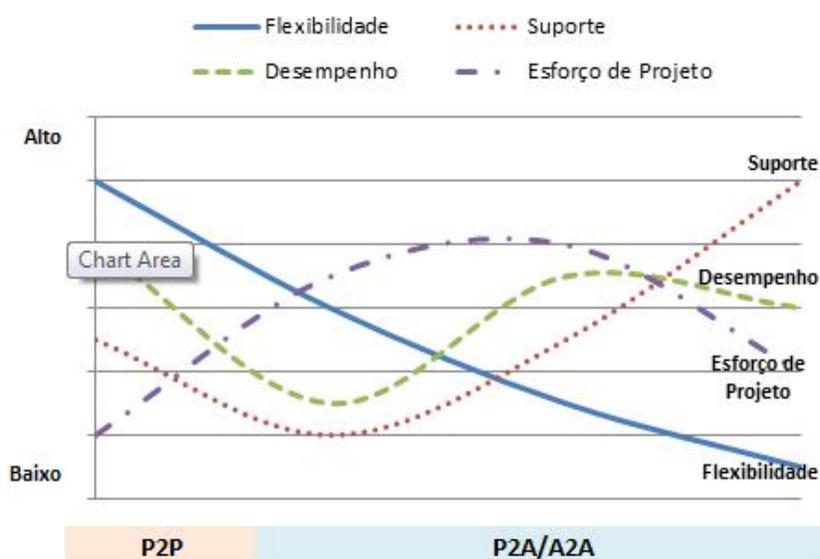


Figura 2.8—Cenário dos tipos de PAIS

Fonte: AALST et al. (2009) adaptado pelo autor

Neste contexto, existe uma grande demanda por parte das organizações, na utilização de PAIS que permitam uma maior flexibilidade no seu ambiente. Esta demanda é decorrente do fato de que “o nível de competitividade de uma empresa é a velocidade com que ela consegue responder às novas demandas, oportunidades e ameaças do mercado” (MAGALHÃES et al. 2004). Porém, para que as organizações possam se adequar rapidamente às mudanças do mercado em que atua, mais flexibilidade nos sistemas de informação de suporte aos seus processos é requerida.

De acordo com Aaslt et al.(2007) um ambiente flexível que possibilite a tomada de decisão de acordo com a demanda atual, envolve os eventos que ocorrem durante a execução dos processos. “A identificação de eventos ao longo do processo, em tempo real, é um importante fator de negócio que corrobora com a

demanda de uma camada de gerenciamento conectada diretamente aos softwares que estão executando o processo” (DE SORDI 2008). Portanto, é necessário que os PAISs possuam uma arquitetura diferenciada para atender as demandas atuais de flexibilidade nos processos de negócio.

“A visualização ou abstração do fluxo de processos de negócio em tempo real, independentemente da quantidade e localização dos softwares que o compõem, o acompanhamento dos indicadores do processo ou de suas partes, o disparo de ações com base em eventos, entre outras funcionalidades requeridas à gestão dos modernos processos colaborativos, requerem uma proposta e arquitetura de software diferente das tradicionais” (DE SORDI 2008).

Com base neste cenário e a fim de permitir mais flexibilidade aos PAISs, garantindo desta forma uma maior capacidade no gerenciamento dos processos de negócio, o presente trabalho utilizará os conceitos de BPM em conjunto com os conceitos de PAIS flexíveis para que seja possível identificar os eventos ao longo da execução dos processos e auxiliar nas decisões dos usuários.

2.3 TEORIA DE CONTROLE DE SUPERVISÓRIO

A evolução das tecnologias de computação, comunicação e sensoriamento permitiram elevar o nível de complexidade de sistemas automatizados. Atualmente, tais sistemas podem ser encontrados em diversas áreas de atuação tais como: nos modernos sistemas de manufatura; na robótica; na supervisão de tráfego; na logística; nas redes de comunicação e de computadores; nos sistemas operacionais; na concepção de software; no gerenciamento de base de dados; na otimização de processos distribuídos; entre outros (HO 1989).

Os sistemas descritos no parágrafo anterior são classificados na literatura como Sistemas a Eventos Discretos (SED). “Um sistema a eventos discreto (SED) é um sistema dinâmico discreto e dirigido a eventos, isto é, com espaço de estados discreto e cuja evolução de estado depende inteiramente da ocorrência assíncrona de eventos discretos.” (CASSANDRAS e LAFORTUNE 1999).

A complexidade destes sistemas gerou a necessidade de um formalismo específico para a representação formal do comportamento dos mesmos bem como para a otimização de seu controle. Nesta seção são apresentados os conceitos fundamentais acerca da formalização e modelagem dos SED. Serão apresentados também os fundamentos da Teoria de Controle Supervisório a qual permite a síntese formal de um agente de controle que coordena de forma minimamente restritiva o comportamento do sistema.

2.3.1 Sistemas a eventos discretos – SED

Segundo Ramadge e Wonham (1987), um SED é um sistema dinâmico que evolui de acordo com a ocorrência abrupta em intervalos de tempos irregulares e desconhecidos dos eventos físicos. Geralmente a ocorrência de um evento causa uma mudança do estado do sistema. O término de uma atividade ou o fim de uma temporização são exemplos de eventos que podem ocorrer no sistema. Tais mudanças se caracterizam por serem abruptas e instantâneas: ao perceber um evento, o sistema reage imediatamente, acomodando-se em tempo nulo em uma nova situação, onde permanece até que ocorra um novo evento (CURY 2001). Desta forma, a simples passagem do tempo não é suficiente para garantir que o sistema evolua, pois, para isso, é necessário que ocorram eventos, sejam estes internos ou externos.

A ocorrência de um evento causa uma transição ou mudança de estado no sistema e o sistema permanece neste estado até que algum outro evento ocorra. A evolução no tempo da ocorrência de eventos pode ser representada pela trajetória percorrida no seu espaço de estados, conforme ilustrado pela figura 2.9.

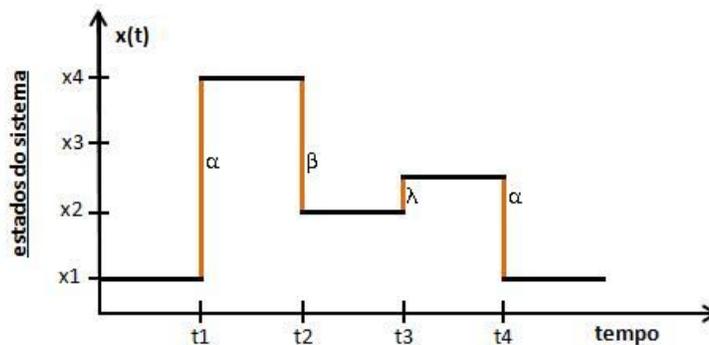


Figura 2.9 - Trajetória típica de um sistema a eventos discretos
 Fonte: Adaptado de Cassandras e Lafortune (1999)

A trajetória ilustrada na figura 2.9 demonstra evolução do comportamento do sistema devido à ocorrência de eventos, os quais estão representados pelos símbolos α , β e λ . É possível identificar que um mesmo evento (no caso da figura acima o evento α) pode causar uma mudança diferente no comportamento (estado) do sistema. Esta diferenciação no comportamento do sistema terá como base o estado atual deste sistema. A figura demonstra que a ocorrência do evento α no estado x_1 modifica o estado do sistema para x_4 . Porém quando este mesmo evento ocorre no estado x_3 , o estado do sistema volta a ser x_1 . A trajetória pode continuar indefinidamente, inclusive com a ocorrência de outros eventos não representados na figura 2.9. Para todos os sistemas tratados, porém, assume-se que o número total de eventos que podem ocorrer é finito. Os sistemas a eventos discretos podem permanecer um tempo arbitrariamente longo em um mesmo estado, sendo que sua trajetória pode ser dada por uma lista de eventos incluindo-se eventualmente os instantes de tempo em que tais eventos ocorreram.

2.3.2 Linguagens Formais

Conforme explanado na seção anterior, os SEDs são conduzidos por eventos, tipicamente representando o início ou fim de uma atividade. Para Cassandras e Lafortune (1999), uma das maneiras formais de estudar o comportamento lógico de um SED é baseada na Teoria de Linguagens. Dado um conjunto de eventos distintos Σ como o alfabeto de um SED, assume-se que Σ seja finito. Entende-se por

Σ^* o conjunto de todas as palavras possíveis formadas pelos eventos constituintes deste alfabeto. Uma seqüência de eventos sobre este alfabeto forma uma "palavra". Uma palavra vazia é representada pelo símbolo ϵ , tal que $\epsilon \notin \Sigma$. Uma linguagem definida sobre um alfabeto Σ é um conjunto de palavras de comprimento finito sobre este alfabeto. Seja, por exemplo, $\Sigma=\{a,b,g\}$ um conjunto de eventos, uma possível linguagem sobre Σ é $L1=\{\epsilon,a,abb\}$, a qual consiste de três palavras. Seja $s \in \Sigma^*$ sendo $s=tuv$ com $t,u,v \in \Sigma^*$, então:

- a) t é chamado de prefixo de s ;
- b) u é chamado de subpalavra de s ; e
- c) v é chamado de sufixo de s .

2.3.3 Operações sobre as linguagens

A literatura define diversas operações sobre linguagens. Dentre elas destacam-se:

- a) O Fechamento Kleene do alfabeto Σ é definido como o conjunto de todas as palavras possíveis, formadas pelos símbolos pertencentes a este alfabeto, incluindo a palavra vazia ϵ , tal conjunto é a linguagem representada por Σ^* ;
- b) A concatenação de duas linguagens $L_a, L_b \subseteq \Sigma^*$, é definida (CARROLL e LONG 1989) como a justaposição de duas linguagens, dando origem a uma linguagem maior, que é composta dos símbolos pertencentes a primeira linguagem, imediatamente seguidos dos símbolos pertencentes a segunda linguagem, e pode ser expressa por:

$$L_a L_b := \{s \in \Sigma^* : (s = s_a s_b) \text{ e } (s_a \in L_a) \text{ e } (s_b \in L_b)\}$$

- c) O prefixo-fechamento de uma linguagem $L \subseteq \Sigma^*$ é definido como o conjunto de todos os prefixos de uma dada linguagem incluindo a palavra vazia ϵ , e pode ser definido por,

$$L := \{s \in \Sigma^* : \exists t \in \Sigma^* (st \in L)\}$$

em geral $L \subseteq L^e$, L é dito ser prefixo-fechada se $L = L^e$.

2.3.4 Teoria de autômatos

Um Autômato Finito Determinístico (RAMADGE e WONHAM, 1987) pode ser descrito da seguinte forma:

$$G=(Q,\Sigma,\delta, Q_m,q_0)$$

sendo:

Q = conjunto de estados;

Σ = o alfabeto de eventos;

$\delta:Q \times \Sigma \rightarrow Q$ = a função de transição de estados;

$Q_m \subseteq Q$ = o conjunto de estados marcados (que representam tarefas executadas);

$q_0 \in Q$ é o estado inicial.

A figura 2.10 (CASSANDRAS e LAFORTUNE 1999) ilustra um grafo com a representação completa de um autômato. Neste grafo os nós representam os estados e os arcos rotulados representam transições entre estes estados. Neste caso $Q=\{x,y,z\}$ e $\Sigma=\{a,b,g\}$. Os arcos no grafo dão uma representação gráfica da função de transição de estados do autômato, que denota-se como $\delta:Q \times \Sigma \rightarrow Q$:

$$\begin{array}{ll} \delta(x,a)= x & \delta(x,g)= z \\ \delta(y,a)= x & \delta(y,b)= y \\ \delta(z,b)= z & \delta(z,a)=f(z,g)= y \end{array}$$

A notação $\delta(y,a)=x$ significa que se o autômato está no estado y , então quando da ocorrência do evento a , o autômato fará uma transição instantânea para o estado x .

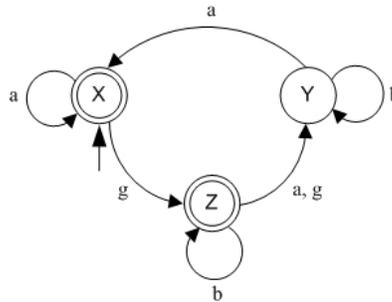


Figura 2.10 - Exemplo de um autômato determinístico
Fonte: Cassandras e Lafortune (1999)

Algumas observações devem ser feitas com relação ao exemplo ilustrado na figura 2.10. Primeiro, um evento pode ocorrer sem causar mudança de estado, como em $\delta(x,a)=x$. Segundo, dois eventos distintos podem ocorrer num dado estado causando a mesma transição como em $\delta(z,a)=\delta(z,g)=y$. Terceiro, a função δ é uma função parcial sobre seu domínio $Q \times \Sigma$, isto é, não é necessário uma transição ser definida para cada evento em Σ para cada estado de Q , por exemplo, $\delta(x,b)$ e $\delta(y,g)$ não são definidas.

O estado inicial é denotado por q_0 . O subconjunto Q_m de Q representa os estados de Q que são marcados. Os estados são marcados quando se deseja associar a este um significado especial, como por exemplo, a conclusão de uma tarefa. Os estados marcados são também referenciados como estados finais.

2.3.5 Linguagens geradas e marcadas

A um autômato são associadas duas linguagens $L(G)$ e $L_m(G)$. A linguagem $L(G)$ representa todas as seqüências de eventos que ela pode gerar, chamada de linguagem gerada, sendo formalmente definida como:

$$L(G) := \{ s \in \Sigma^* : \delta(q_0, s) \text{ está definida} \}$$

$L_m(G)$ representa o comportamento marcado, ou seja, tarefas completas executadas pela planta chamada de linguagem marcada, sendo a sua definição como segue abaixo:

$$L_m(G) := \{ s \in L(G) : \delta_{q_0, s} \in Q_m \},$$

estas duas linguagens são executadas sobre o alfabeto Σ e fazem parte do conjunto Σ^* . Diz-se que um autômato é não-bloqueante quando $L_m(G) = L(G)$.

Sendo $\Sigma = \{a, b\}$ um conjunto de eventos, considere a linguagem:

$$L = \{a, aa, ba, aaa, aba, baa, bba, \dots\}$$

consistindo de todas as palavras de a e b sempre seguidas por um evento a. Esta linguagem é marcada pelo autômato de estado finito $G = \{Q, \Sigma, \delta, Q_m, q_0\}$ onde $Q = \{0, 1\}$, $q_0 = 0$, $Q_m = \{1\}$ e δ é definido como: $\delta(0, a) = 1$, $\delta(0, b) = 0$, $\delta(1, a) = 1$, $\delta(1, b) = 0$.

No autômato apresentado na figura 2.11 têm-se o estado 0 (zero) como estado inicial e o estado 1 (um) como estado marcado. Verifica-se que $L_m(G) = L$. Nota-se que δ é uma função total, ou seja, para todo evento de Σ é definida uma transição em cada estado de Q e, além disso, a linguagem gerada por G é $L(G) = \Sigma^*$.

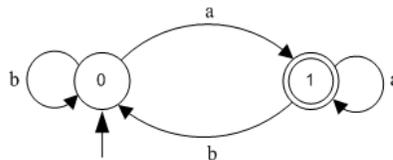


Figura 2.11 - Autômato que marca a linguagem consistindo de todas as palavras de **a** e **b**
Fonte: Cassandras e Lafortune (1999)

Ao remover-se do autômato, ilustrado na figura 2.11, o auto-circo referente ao evento b no estado 0 (zero), isto é, tornar $\delta(0, b)$ indefinida, então $L(G)$ agora consiste daquelas palavras em Σ^* que iniciam com o evento a e nas quais não existe a ocorrência consecutiva de b. Neste caso, ou o evento b é o último evento da palavra, ou ele será seguido imediatamente por um evento a.

2.3.6 Operações sobre autômatos

Ao estudar os SEDs modelados por Autômatos, necessita-se de um conjunto de operações que permitam combinar ou compor dois ou mais autômatos, bem como analisar as propriedades destes. No autômato da figura 2.12 (CURY 2001) pode-se atentar para dois estados em particular, X_1 e X_5 , em função dos quais serão definidos os conceitos de acessibilidade e co-acessibilidade respectivamente:

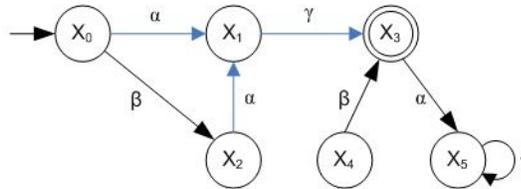


Figura 2.12 - Autômato não-acessível e não co-acessível
Fonte: Cury (2001)

X_1 : estado não-acessível

X_5 : estado não co-acessível

Um estado é acessível, se este pode ser atingido ou alcançado através de qualquer cadeia $s \in L(G)$ tendo como origem o estado $q_0 \in Q$. Das definições de $L(G)$ e $L_m(G)$, observa-se que é possível apagar de G todos os estados que não são acessíveis ou alcançáveis a partir de q_0 através de alguma palavra de $L(G)$, sem afetar as linguagens geradas e marcadas por G . Um autômato acessível é aquele em que todos os estados podem ser alcançados a partir do estado inicial q_0 . A figura 2.13 ilustra um exemplo de autômato acessível.

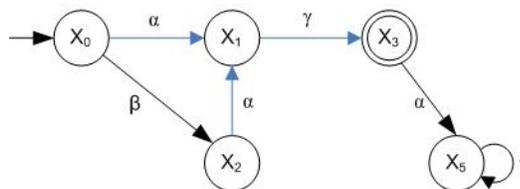


Figura 2.13 - Autômato acessível
Fonte: Cassandras e Lafortune (1999)

Quando se apaga um estado que não é acessível, também são excluídas todas as transições ligadas àquele estado. Chama-se esta operação por $Ac(G)$, onde Ac representa a parte acessível. Formalmente,

$$\begin{aligned}
 \text{Ac}(G) &:= (Q_{\text{ac}}, \Sigma, \bar{\delta}_{\text{ac}}, q_0, Q_{\text{ac},m}) \\
 Q_{\text{ac}} &= \{q \in Q : \exists s \in \Sigma^* \bar{\delta} q_0, s = q\} \\
 Q_{\text{ac},m} &= Q_m \cap Q_{\text{ac}} \\
 \bar{\delta}_{\text{ac}} &= \bar{\delta}|_{Q_{\text{ac}} \times \Sigma \rightarrow Q_{\text{ac}}}
 \end{aligned}$$

Um estado q de G é dito ser co-acessível para Q_m , se há uma cadeia em Σ^* com início neste estado q e final em um estado $q \in Q_m$. No exemplo anterior tem-se que o estado X_5 é não co-acessível, já que não existe um caminho de X_5 para um estado marcado. Denota-se a operação de apagar todos os estados de G que não são co-acessíveis por $\text{CoAc}(G)$.

A operação CoAc pode alterar $L(G)$, visto que podem ser apagados estados que são acessíveis. Contudo a operação CoAc não afeta $L_m(G)$. Se $G = \text{CoAc}(G)$, então G é dito ser co-acessível, neste caso $L(G) = L_m(G)$.

Um autômato acessível e co-acessível ao mesmo tempo é dito Trim. Um autômato Trim representa a ausência de bloqueios no sistema, isto é, a partir do estado inicial q_0 ou de qualquer outro estado, sempre existirá um caminho que conduz a um estado marcado. A operação Trim é definida como segue:

$$\text{Trim}(G) := \text{CoAc}[\text{Ac}(G)] = \text{Ac}[\text{CoAc}(G)]$$

O comportamento da operação Trim pode ser observado na figura 2.14.

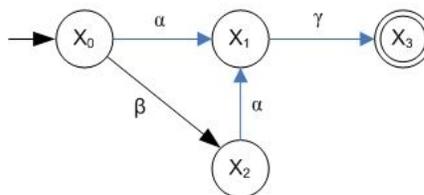


Figura 2.14 - Autômato Trim
Fonte: Cassandras e Lafortune (1999)

A composição paralela (ou síncrona) é uma operação que representa o comportamento sincronizado entre dois autômatos. Nesta operação um evento comum aos dois alfabetos de eventos, é permitido ocorrer se e somente se ele ocorre em ambos os autômatos simultaneamente. No caso de um evento pertencer

a somente um dos alfabetos, ele sempre estará habilitado a ocorrer no autômato resultante quando se realizar a composição. Desta forma o sincronismo se observa somente quando são analisados eventos comuns a ambos os alfabetos.

Um estado do autômato composto será marcado se ele resultar de estados marcados em ambos os geradores. Sejam os autômatos $G_1=(Q_1, \Sigma_1, q_{01}, Q_{m1}, \delta_1)$ e $G_2=(Q_2, \Sigma_2, q_{02}, Q_{m2}, \delta_2)$, define-se a composição paralela de G_1 e G_2 , denotada $G_1 \parallel G_2$, pelo autômato

$$G_1 \parallel G_2 := Ac(Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2, q_{01}, q_{02}, Q_{m1} \times Q_{m2}, \delta_{12})$$

onde $\delta_{12}: (Q_1 \times Q_2) \times (\Sigma_1 \cup \Sigma_2) \rightarrow (Q_1 \times Q_2)$ é definida por

$$\delta_{12}(q_1, q_2, \sigma) = \begin{cases} \delta_1(q_1, \sigma), \delta_2(q_2, \sigma) & \text{se } \sigma \in \Sigma_{G_1}(q_1) \cap \Sigma_{G_2}(q_2) \\ \delta_1(q_1, \sigma), q_2 & \text{se } \sigma \in \Sigma_{G_1} - \Sigma_2 \\ q_1, \delta_2(q_2, \sigma) & \text{se } \sigma \in \Sigma_{G_2} - \Sigma_1 \\ \text{não definida} & \text{caso contrário} \end{cases}$$

para $(q_1, q_2) \in Q_1 \times Q_2$ e $\sigma \in \Sigma_1 \cup \Sigma_2$ (CURY 2001).

Sejam os autômatos G_1 e G_2 representativos de um processo qualquer, sendo o estado inicial 0 (zero) significando o estado de repouso e o estado 1 (um), o estado representativo do sistema em operação, os eventos (α_1, α_2) representam início de operação, e os eventos (β_1, β_2) significam final de operação. A figura 2.15 ilustra os dois autômatos e os eventos de início e fim de operação e a figura 2.16 ilustra o autômato resultante da composição paralela dos autômatos G_1 e G_2 .

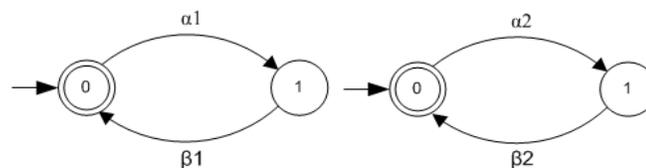


Figura 2.15 – Autômatos G_1 e G_2
Fonte: Cassandras e Lafortune (1999)

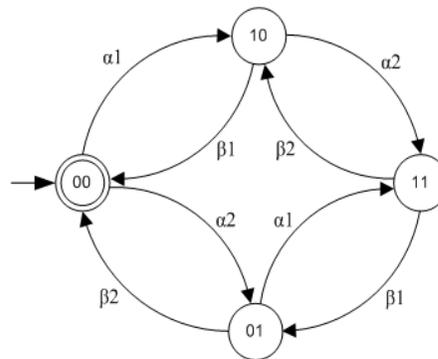


Figura 2.16 - Autômato $G_1 \parallel G_2$
 Fonte: Cassandras e Lafortune (1999)

Observa-se na figura 2.16 que em função dos autômatos G_1 e G_2 serem assíncronos, ou seja, não possuírem eventos em comum, a evolução do autômato resultante determina que todos os eventos definidos em ambos os autômatos também estarão definidos em $(Q_1 \times Q_2)$. Por exemplo, o estado 01 representa a sincronização do estado 0 (zero) de G_1 , com o estado 1 (um) de G_2 . Isto significa que os eventos α_1 e β_2 deverão estar definidos.

2.3.7 Supervisão centralizada (Controle Monolítico)

O objetivo do controle supervisorio monolítico é projetar um agente de controle único, usualmente denominado supervisor, cuja função é restringir a evolução do sistema através da desabilitação da ocorrência de determinados eventos conforme a seqüência de eventos observados na planta. O projeto do supervisor é realizado com base em um conjunto de especificações ou regras comportamentais. Desta forma o comportamento do sistema em malha fechada (sob a ação do agente de controle) é restringido por tais regras operacionais especificadas (QUEIROZ e CURY, 2000).

De modo a modelar um sistema de controle o conjunto de eventos Σ é dividido em dois conjuntos, *eventos controláveis* (Σ_c) e *eventos não-controláveis* (Σ_u), de forma que $\Sigma = \Sigma_c \cup \Sigma_u$. Os eventos Σ_c podem ser desabilitados a qualquer momento pelo supervisor, ao passo que os eventos Σ_u não são afetados pela ação do supervisor.

Seja $\Delta = \{0,1\}^{\Sigma_c}$ o conjunto de todas as habilitações binárias para os elementos de Σ_c . Cada habilitação $\gamma \in \Delta$, ou seja, cada função

$$\gamma: \Sigma_c \rightarrow \{0,1\}$$

é um *padrão de controle*.

Um evento σ é dito estar habilitado por γ se $\gamma(\sigma)=1$ ou desabilitado por γ se $\gamma(\sigma)=0$. O subconjunto de eventos (γ) que estão aptos a ocorrer em qualquer momento, na operação da planta, é definido através da determinação das entradas de controle. Uma entrada de controle para o gerador G consiste do subconjunto $\gamma \subseteq \Sigma$ que satisfaça $\Sigma_u \subseteq \gamma$. Esta condição significa que qualquer evento não-controlável sempre será permitido ocorrer, ou seja, não está sujeito à ação de inabilitação por parte do supervisor (RAMADGE e WONHAM 1989).

O conjunto das entradas de controle é denotado por $\Psi \subseteq 2^{\Sigma}$. Formalmente (RAMADGE e WONHAM 1989) um supervisor é um mapa que corresponde a seguinte função $f: L(G) \rightarrow \Psi$, que associa a cada cadeia $s \in L(G)$ gerada pela planta, uma entrada de controle $\gamma = f(s)$ que restringe o comportamento da planta até a ocorrência de um novo evento.

Quando um SED (RAMADGE e WONHAM 1989) está sob a ação de controle de um supervisor S , ele agora obedece às restrições adicionais. Após a geração de uma palavra w pela planta, o próximo evento deve ser um elemento de $f(w) \cup \Sigma(\delta(w, q_0))$. Define-se $\Sigma(q)$ como conjunto de eventos factíveis no estado q com $\Sigma(q) \subseteq \Sigma$, de modo que para cada $\sigma \in \Sigma(q)$, $\delta(\sigma, q)$ é definida. Portanto o evento a ser gerado após a palavra w deverá ter simultaneamente a sua entrada de controle habilitada pelo supervisor e a função de transição definida na planta satisfazendo $f(w) \cap \Sigma(\delta(w, q_0))$.

O comportamento do sistema composto pela planta G , sujeito à ação do supervisor S , é denotado por $(\frac{S}{G})$. Seja $L(\frac{S}{G})$ a linguagem que representa o comportamento do sistema sob supervisão, sendo $L(\frac{S}{G}) \subseteq L(G)$ definida formalmente como segue:

$$L(\frac{S}{G}) \subseteq L(G) = \begin{cases} \epsilon \in L(\frac{S}{G}) \\ w\sigma \in L(\frac{S}{G}) \text{ se e somente se } w \in L(\frac{S}{G}), \sigma \in f(w) \text{ e } w\sigma \in L(G) \end{cases}$$

A linguagem marcada $L_m(\frac{S}{G})$ que representa o comportamento marcado do sistema sob supervisão é definida como:

$$L_m \frac{S}{G} = L_m(G) \cap L(\frac{S}{G})$$

Ou seja, ela representa o conjunto de tarefas completas realizadas pela planta e que sobreviveram à ação de controle do supervisor.

Um supervisor S é dito ser não-bloqueante para um gerador G , se $L_m(\frac{S}{G}) = L(\frac{S}{G})$. Isto implica que, as palavras ou seqüências de eventos geradas pelo sistema sob supervisão sempre deverão evoluir no sentido de concluir-se uma tarefa imposta pelo sistema.

O supervisor que afeta o comportamento da planta ao habilitar e desabilitar eventos controláveis pode ser descrito, ou representado na prática, por um autômato S e uma função $\Phi: X \rightarrow \psi$, onde $S = (X, \Sigma, \xi, X_m, x_0)$, com: X é o conjunto de estados x , Σ é o alfabeto utilizado por G , ξ é função de transição, X_m é o conjunto de estados marcados e x_0 é o estado inicial. Seja $\Phi: X \rightarrow \psi$ a função que mapeia os estados de S em entradas de controle para G , diz-se que o par (S, Φ) executa o supervisor S se para cada $w \in L(S, G)$.

$$\Phi \xi(w, x_0) = f(w)$$

Isto significa que o valor de S para uma dada seqüência de eventos w gerada pela planta G é obtido aplicando-se w ao autômato S , a partir do seu estado inicial x_0 , obtendo-se um novo estado x' , cuja função de transição é definida por $x' = \xi(w, x_0)$. Assim a função Φ fornece a entrada de controle que deverá ser aplicada a seqüência de eventos w , ou seja, $f(w) = \gamma = \Phi(x')$.

De modo prático, o supervisor S será representado graficamente pelo autômato S , onde as ações de controle de S sobre a planta G estarão definidas na estrutura de transição de S . Os eventos não habilitados pelo supervisor S não aparecerão na estrutura do autômato S , assim, na operação de composição

síncrona, como os alfabetos são comuns, um evento só poderá ocorrer, se ele ocorrer simultaneamente em ambos os autômatos de S e G.

Desta forma, dado o autômato S que representa S, se uma dada cadeia $s \in L(S/G)$, então $s \in L(S)$ e $s \in L(G)$ somente se $\sigma \in f(s)$. Ainda assim, se $s \in L(S/G)$, $s \in L(G)$ e $\sigma \in f(s)$, então $s \in L(S)$. A primeira afirmação assegura que as transições desabilitadas por S, não aparecem na estrutura de transição de S, ao passo que a segunda, assegura que as transições habilitadas por S, e possível de ocorrer em G, aparecerão na estrutura de transição de S.

Na figura 2.17 é ilustrada a estrutura de um sistema de controle a eventos discretos com supervisão monolítica. Observa-se que a planta gera os eventos controláveis e não-controláveis, cabendo ao supervisor a ação de desabilitar somente os eventos controláveis.



Figura 2.17 - Esquema do controle monolítico

A figura 2.17 apresenta uma Arquitetura de Controle Supervisório (SCA). A SCA segundo Ramadge e Wonham (1989) define que a planta (sistema a ser controlado) gera espontaneamente todos os eventos e é papel dos supervisores habilitar/desabilitar os eventos controláveis. Desta forma, todos os eventos em Σ executados pela planta G são observados pelo supervisor S. Na figura 2.17, w é a seqüência de todos os eventos executados por G e que são observados por S.

A SCA apresentada na figura 2.17 é a base para a definição do Serviço de Controle Supervisório (SCS), conforme será apresentado nos próximos capítulos. O supervisor obtido utilizando a Teoria de Controle Supervisório é a principal entidade implementada no serviço de controle de supervisão. No entanto, a fim de implementar o sistema de controle sob o enfoque desta teoria é necessário lidar com algumas limitações do modelo de Ramadge e Wonham (1989). A complexidade da síntese de supervisores, embora polinomial no número de estados da planta e

modelos de especificação, é um obstáculo em aplicações onde o número de estados que representam o sistema aumenta exponencialmente de acordo com o número de seus componentes. Este fato é especialmente relevante para sistemas de grande escala, tendo sido considerado por diversos autores que tentam superar as dificuldades computacionais explorando diferentes aspectos do sistema, em particular, destaca-se a abordagem Modular Local (LMC) (QUEIROZ e CURY 2000).

2.3.8 Abordagem Modular Local da Teoria de Controle Supervisório

De acordo com Queiroz e Cury (2002), o projeto de SEDs envolve, tipicamente, uma grande quantidade de especificações comportamentais, desta forma, a abordagem de controle modular é freqüentemente usada para a síntese de controladores. Ao invés de se projetar um único supervisor monolítico que satisfaça todas as especificações, procura-se construir um supervisor para cada especificação individual. Neste caso, deseja-se que os supervisores resultantes sejam modulares, isto é, que a ação conjunta dos supervisores modulares tenha o mesmo desempenho que a de um supervisor monolítico equivalente. Quando esta propriedade é verificada, a abordagem de controle modular é bastante vantajosa no sentido de promover maior flexibilidade, maior eficiência computacional e segurança na aplicação do controle.

Um funcionamento conjunto dos supervisores localmente modulares é ilustrado na figura 2.18. Nesta figura é possível verificar que os supervisores locais observam e controlam apenas o comportamento dos subsistemas afetados pela especificação correspondente. Assim, pode-se afirmar que a abordagem de controle modular local induz uma estrutura de controle descentralizada que surge naturalmente do processo de síntese.



Figura 2.18 - Esquema de controle modular local

De acordo com a abordagem LMC (QUEIROZ e CURY 2000), o sistema a ser controlado é modelado por uma Representação por Sistema Produto (PSR), ou seja, por um conjunto de subsistemas assíncronos $G_i | i \in I$ tal que todos os pares de subsistemas neste conjunto têm alfabetos distintos, isto é, não compartilham eventos comuns (RAMADGE e WONHAM 1989). O comportamento de cada subsistema é representado por um autômato $G_i = (\Sigma^{G_i}, Q^{G_i}, \delta^{G_i}, q_0^{G_i}, Q_m^{G_i})$, de modo que o comportamento de todo o sistema a ser controlado é obtido pelo produto síncrono de todos os subsistemas da PSR, por exemplo, $G = \prod_{i \in I} G_i$. O conjunto de eventos é $\Sigma = \cup_{i \in I} \Sigma^{G_i}$. Considerando-se um subsistema em $G_i | i \in I, \Sigma_c^{G_i}$ denota seu conjunto de eventos controláveis e $\Sigma_{uc}^{G_i}$ seu conjunto de eventos não-controláveis. Além disso, sistemas de grande escala são caracterizados por terem diversas especificações, cada uma atuando em apenas parte do sistema global. Geralmente, essas especificações tentam sincronizar alguns subsistemas concorrentes. Esta peculiaridade da PSR é explorada pelo controle modular local a seguir.

Na abordagem modular local, ao invés de sintetizar um supervisor global único que satisfaz todo o conjunto de especificações, um supervisor local é sintetizado de forma a satisfazer cada especificação. Isto significa que cada um dos supervisores locais restringe o comportamento de uma parte do sistema a ser controlado. Esta parte é a planta local correspondente ao supervisor considerado. A planta local G_j é obtida através da realização do produto síncrono dos subsistemas que compartilham eventos com base na especificação considerada. A síntese de um supervisor local S_j é realizada considerando a especificação correspondente $E_j (\Sigma_j \subseteq \Sigma)$ e suas plantas locais $G_i (G_j = \prod_{i \in N_j} G_i)$, com $N_j = \{k \in N | \Sigma_k \cap \Sigma_j \neq \emptyset\}$. Assim, a planta local G_j junta apenas os subsistemas do sistema original composto que estão diretamente restringidos por E_j . Ao utilizar este procedimento, é possível sintetizar

um supervisor local para cada uma das especificações estabelecidas. Se pelo menos um supervisor local no conjunto $S_{ij} \in J$ desativa a ocorrência de um evento então a ocorrência desse evento é desativada em G . Mesmo quando todos os supervisores locais são não-bloquantes, a ação de controle conjunta pode resultar no bloqueio do sistema. Assim, após realizar o procedimento de síntese, é necessário verificar a propriedade da modularidade do conjunto de supervisores como indicado em (Queiroz e Cury, 2000). Quando essa condição é satisfeita, é possível assegurar que a abordagem modular não causa nenhuma perda de desempenho com relação à abordagem monolítica.

2.4 CONCLUSÃO DO CAPÍTULO

Neste capítulo foram descritos os principais conceitos para o desenvolvimento da presente pesquisa. Inicialmente foram conceituados a Gestão de Processos Negócio (BPM), e apresentadas as principais características e técnicas que envolvem estes conceitos. Ao analisar os conceitos descritos, entende-se que o BPM reflete uma abordagem administrativa que pode utilizar as técnicas e ferramentas da tecnologia da informação para implementação de melhores práticas. O BPM caracteriza-se pela grande utilização e adoção de tecnologias como suporte ao gerenciamento de todo o ciclo de vida dos processos.

Ainda neste capítulo foi fundamentado o conceito dos Process-Aware Information Systems (PAIS). Os PAISs são considerados sistemas para gerenciamento de processos de negócio. Na seção que descreve o PAIS destacou-se a atual necessidade das organizações de possuírem sistemas flexíveis para suporte aos processos de negócio. Para o desenvolvimento da presente pesquisa foram selecionadas duas ferramentas PAIS: o YAWL System e o BIZAGI. Como mencionado o YAWL é uma notação baseada em Redes de Petri que permite a modelagem e especificação de processos de negócio. Além da linguagem, existe também o sistema YAWL System que é uma ferramenta do tipo: BPMS. Por meio do YAWL System é possível realizar todo o ciclo de BPM para os processos de negócio. Assim como o YAWL System, o Bizagi também é uma ferramenta do tipo BPMS. Porém a notação utilizada pelo Bizagi para a modelagem dos processos é o

BPMN. Além disso, o Bizagi possui uma característica mais comercial, visto que a versão que permite a realização de todo o ciclo do BPM é paga.

Por fim, foram apresentados os principais conceitos da Teoria de Controle de Supervisório. Estes conceitos são a base para o desenvolvimento da proposta desta pesquisa: a construção de um sistema de controle e supervisão – Serviço de Controle Supervisório (SCS). O SCS proposto utilizará em sua estrutura a teoria dos autômatos para possibilitar a especificação das regras de negócio dos processos e analisar as regras de negócio especificadas para recomendar ao usuário os próximos passos, além da arquitetura orientada a serviços para possibilitar a comunicação entre os PAISs e o SCS.

3 ABORDAGEM DE CONTROLE UTILIZADA NO SCS

Este capítulo apresenta uma abordagem formal para a implementação dos resultados teóricos obtidos a partir da teoria de controle supervísório. Inicialmente é apresentada uma visão geral do funcionamento do SCS e o ciclo de desenvolvimento do sistema de supervisão. Com base neste ciclo, é descrita a arquitetura que será utilizada para o desenvolvimento do SCS, a Arquitetura do Serviço de Controle Supervísório (SCSA). A arquitetura define a estrutura lógica a ser implementada no SCS para supervisionar o PAIS e possibilitar a utilização de um serviço de recomendação para guiar os usuários na execução das atividades do processo.

3.1 VISÃO GERAL DO SCS

De acordo com a interpretação usualmente adotada na Teoria de Controle Supervísório o sistema a ser controlado é constituído por diversos subsistemas elementares. Quando este é um sistema de manufatura, cada subsistema corresponde a um equipamento que opera de forma independente dos demais. O comportamento de cada um dos subsistemas é representado através de um autômato. O comportamento do sistema global, sem qualquer restrição comportamental, é obtido realizando o produto síncrono dos autômatos associados a cada um destes subsistemas. O comportamento concorrente destes subsistemas deve ser restringido de forma a evitar que situações consideradas indesejáveis ocorram, como por exemplo, a colisão entre robôs manipuladores ou quando a capacidade de um armazém não é respeitada. O comportamento destes subsistemas também deve ser coordenado de forma a garantir que o mesmo satisfaça sua função elementar, ou seja, produzir determinado produto numa quantidade pré-estabelecida.

Neste trabalho a aplicação desta teoria para a coordenação de processos de negócio pressupõe que o sistema a ser controlado é o processo de negócio como um todo e que cada atividade deste processo corresponde a um de seus

subsistemas constituintes. As especificações de controle são estabelecidas de forma a coordenar a seqüência de execução das atividades. A figura 3.1 ilustra um modelo para representar uma determinada atividade de um processo de negócio. Na tabela 3.1 é apresentado o significado de cada estado e cada evento deste modelo.

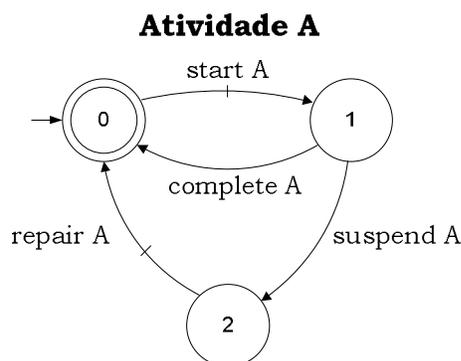


Figura 3.1 – Modelo para representação de uma atividade constituinte de um processo de negócio

Tabela 3.1 – Semântica de estados e eventos empregados no modelo da Figura 3.1

Elemento	Definição
Estado: 0	O estado 0 indica o estado inicial da atividade A antes de sua execução ou ao final da sua execução. Enquanto a atividade A está neste estado, pode existir a ocorrência do evento: start_A.
Evento: start_A	O evento start_A ocorre no estado inicial da atividade A. A partir deste evento, a atividade A alcança o estado 1.
Estado: 1	O estado 1 indica que a atividade A está em execução. Neste estado, os seguintes eventos podem ocorrer: complete_A e suspend_A.
Evento: complete_A	O evento complete_A ocorre quando a atividade A está em execução e indica a finalização da execução da atividade. A partir deste evento, a atividade A alcança o estado inicial novamente, o estado 0.
Evento: suspend_A	O evento suspend_A ocorre quando a atividade A está em execução e indica que ocorreu alguma falha durante a execução da atividade. A partir deste evento, a atividade A alcança o estado 2.
Estado: 2	O estado 2 indica que a atividade A está aguardando a correção da falha que ocorreu durante a sua execução. Neste estado, o seguinte evento pode ocorrer: repair_A.
Evento: repair_A	O evento repair_A ocorre quando a atividade A está aguardando a correção da falha que ocorreu durante a execução da atividade. A partir deste evento, a atividade A alcança o estado inicial novamente, o estado 0.

A figura 3.2 ilustra uma visão geral da Arquitetura de Controle Supervisório. Em geral, cada processo de negócio a ser suportado pelo SCS deve ser descrito como um modelo de processo, e após o modelo validado, deve ser implementado e executado com o suporte de um PAIS. O SCS pode ser utilizado quando o fluxo do processo fornece aos usuários flexibilidade na execução das atividades, mesmo considerando as limitações impostas pelo modelo e regras de negócio do processo.

Um exemplo para a utilização do SCS ocorre quando múltiplas atividades podem ser executadas simultaneamente. Neste caso, os usuários podem definir quais das atividades habilitadas serão executadas.

Em geral, é esperado dos usuários a definição da melhor seqüência para a execução das atividades a fim de alcançar os objetivos estipulados para o processo. Contudo, considerando a falta de informação ou mesmo de conhecimento acerca do processo e seu impacto nas diretrizes organizacionais, muitas vezes a escolha pelos usuários pode não ser a mais adequada. Assim, algumas dessas decisões podem ter uma conseqüência indesejada, no sentido de que elas podem violar algumas regras estabelecidas ou reduzir o desempenho do processo relacionado.

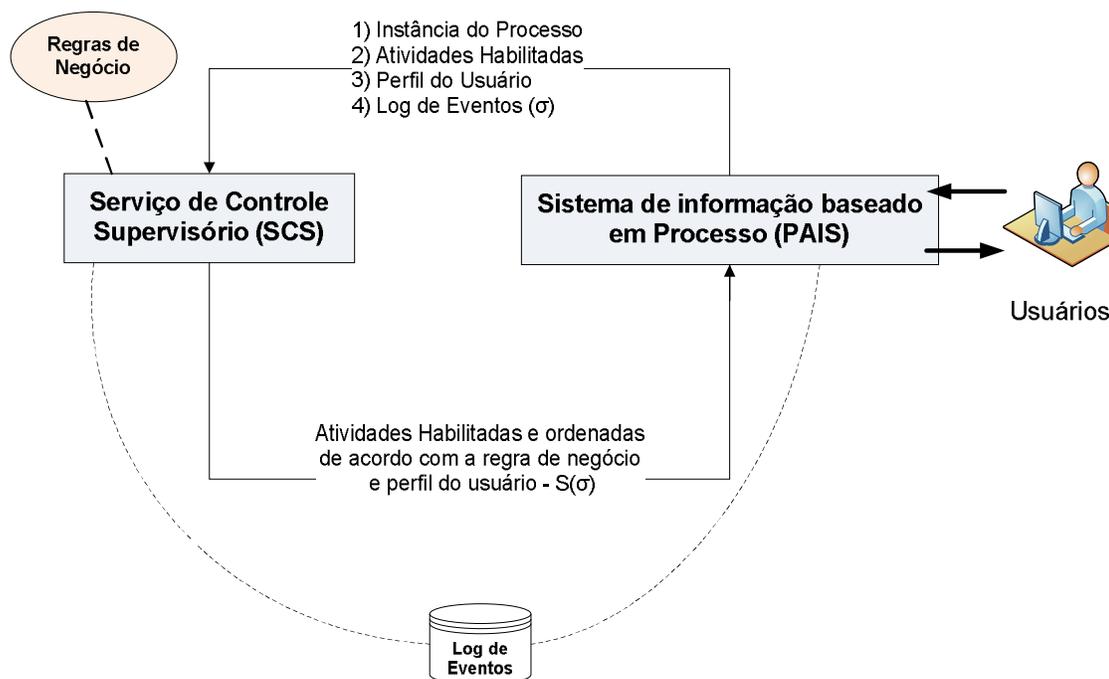


Figura 3.2–Visão Geral da Arquitetura de Controle Supervisorio

Visando prover auxílio aos usuários, o SCS poderá informar e/ou, dependendo do perfil do usuário, restringir a realização das atividades durante a execução do processo. As regras de negócio ou as especificações são definidas de modo a estabelecer o comportamento do SCS e conseqüentemente o suporte na execução do processo. Desta forma, o SCS terá como guiar os usuários na execução das atividades com base nas metas ou regras de negócio definidas.

Considerando a Teoria de Controle Supervisório, o SCS atuará como o supervisor e o PAIS como o sistema a ser controlado.

A fim de implementar o SCS, alguns requisitos são necessários. Em primeiro lugar, deve ser utilizado um PAIS que possa gerar um registro de eventos. O PAIS deve registrar as informações sobre as atividades realizadas nos registros de eventos. Normalmente, os registros de eventos contêm informações sobre o início e conclusão das atividades, a sua ordenação, os recursos que realizaram as atividades e as instâncias do processo a que pertencem (AALST et al. 2003).

Diferente da análise de conformidade (ROZINAT e AALST 2008), onde supõe-se que uma tarefa está associada a, no máximo, um tipo de registro de evento (normalmente um evento completo), a abordagem utilizada pela presente pesquisa considera o registro de evento das atividades de maneira mais abrangente. Por exemplo, o início, a conclusão e a suspensão de uma atividade são eventos considerados pelo SCS. Em segundo lugar, pressupõe-se que o SCS trabalha em modo *'on line'*, ou seja, os eventos são observados pelo supervisor de forma ordenada e não ambígua (considera-se também que a dinâmica da planta é lenta em relação ao sistema de controle). Desta forma, o PAIS deve estar equipado com um mecanismo de registro de eventos que permita a visualização e/ou integração do SCS. Atualmente, os sistemas de informação podem oferecer esse mecanismo de alguma forma, no entanto, esta é uma suposição que qualquer PAIS tenha como recurso o registro de eventos. Além disso, uma infra-estrutura de comunicação é necessária para suportar a conexão e integração entre o PAIS e o SCS. Neste trabalho está sendo considerada a utilização de SOA para este objetivo. Em terceiro lugar, um formalismo bem consolidado para construir o motor do SCS é fundamental. Como os PAISs estão cada vez mais complexos, em muitas situações, é preciso lidar com um grande conjunto de regras (complexas). A consequência é o aumento da complexidade do SCS para ser construído. Assim, o formalismo utilizado para construir o SCS tem de lidar com questões como bloqueio, otimização (qual é a melhor solução?), a viabilidade de implementação, e assim por diante.

Como ilustrado na figura 3.2, o SCS funciona da seguinte forma. O motor do PAIS gera um conjunto de elementos de informação que é completamente observado e tratado pelo SCS. Este conjunto é formado por uma seqüência de caracteres T (uma seqüência de eventos), uma lista de atividades habilitadas (após T), uma instância do processo e o perfil do usuário. De acordo com o controle de

regras estabelecidas no SCS e com o perfil do usuário, um guia com uma lista de atividades habilitadas é retornado ao PAIS. Quando o usuário possui perfil de operador, haverá restrição na execução das atividades. Esta restrição ocorre para as atividades que estão desabilitadas de acordo com as regras de negócio e eventos gerados, inviabilizando desta forma a ocorrência de violação das regras de negócio por meio de usuários menos experientes no processo. Assim, o SCS fornece um "mapa" ou "guia" para os usuários, informando-lhes quais as atividades que podem ser executadas e quais as atividades que não podem ser executadas ou que não deveriam ser executadas com base nas regras de negócio previamente definidas.

Considerando que o SCS sempre permite alcançar um estado marcado (ou seja, concluir todas as tarefas estabelecidas), os usuários podem optar por realizar uma atividade na lista de atividades habilitadas. Com o aumento da complexidade nos processos flexíveis, na vida real, há tantas opções para os usuários, que um suporte ao usuário se torna fundamental. Restringir ou apresentar essas opções com base em regras de controle pode garantir a eficiência, assertividade e apoio. Ao mesmo tempo, como o SCS permite facilmente e de forma sistemática, a alteração das regras de negócio, esta abordagem é declarativa tornando-a muito flexível e personalizável. Por exemplo, se alguma regra de controle deve ser modificada, incluída ou excluída, é fácil sintetizar um novo controlador e atualizar o SCS.

3.2 CICLO DE DESENVOLVIMENTO DE CONTROLE E SUPERVISÃO

O ciclo de desenvolvimento de controle e supervisão envolve tipicamente as seguintes etapas: Modelagem do processo de negócio (das atividades e especificações), Síntese dos supervisores, Implementação, Validação e Testes, conforme demonstra a figura 3.3. As etapas são desenvolvidas de forma contínua e cíclica. Esta forma de desenvolvimento permite uma revisão contínua dos resultados obtidos em cada etapa e garante a flexibilidade na implementação de novas regras por meio da reconfiguração do sistema. Quando ocorrerem mudanças no sistema supervisionado, tais como a necessidade de uma reconfiguração de processos ou mesmo devido a modificação das regras de negócio, todas as etapas

descritas acima serão novamente executadas a fim de permitir a construção de um novo sistema de controle que irá controlar adequadamente esta nova aplicação.

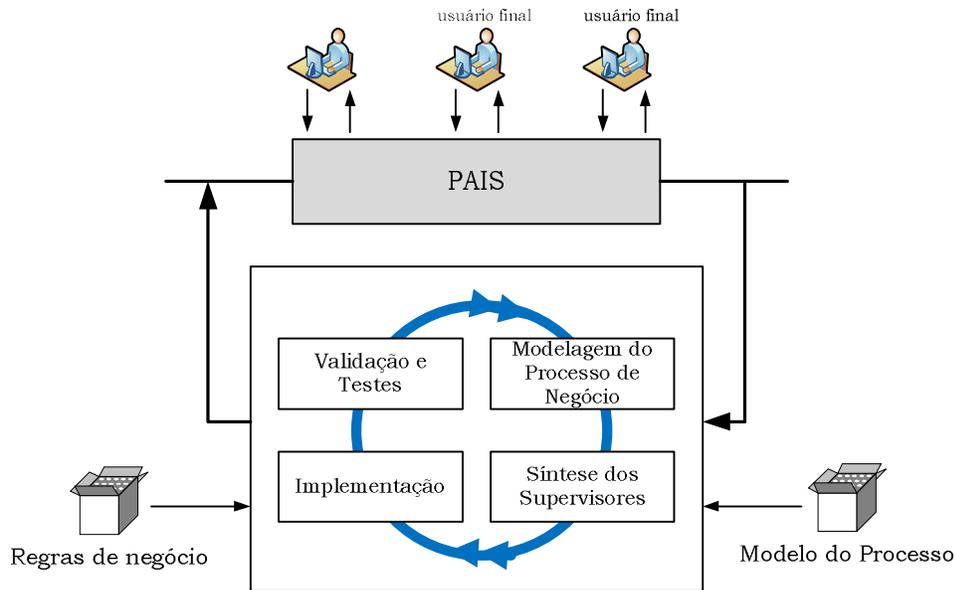


Figura 3.3 - Ciclo de Desenvolvimento de Controle e Supervisão

Na etapa de modelagem serão identificadas as características, necessidades e regras de negócio do sistema que será supervisionado. Sugere-se a divisão desta etapa em duas fases:

- a) modelagem do sistema a ser controlado: nesta fase serão identificadas e representadas as diversas atividades do processo de negócio e os eventos influentes sobre as mesmas (início, término com sucesso, término com falha, pausa, alocação de responsável, ...)
- b) modelagem das regras de negócio do sistema a ser controlado: nesta fase serão identificadas e representadas as diversas restrições entre as atividades (evitar alocar duas atividades em específico para um mesmo usuário, evitar que uma atividade seja iniciada antes que outra tenha sido concluída, ...)

Na etapa de modelagem será importante a participação de um especialista do negócio bem como um especialista na Teoria de Controle Supervisório, para que as atividades e regras de negócio sejam corretamente traduzidas na forma de

autômatos. Destaca-se que, a modelagem do sistema a ser controlado e das regras de negócio são mutuamente dependentes e influentes, devendo ser realizadas de forma iterativa.

Na etapa de implementação será realizada a construção e/ou atualização do sistema de controle e supervisão, e as regras de controle especificadas deverão ser incorporadas ao sistema. Também será realizada a integração do sistema a ser supervisionado com o sistema de controle e supervisão. Esta implementação será realizada utilizando as tecnologias computacionais como: linguagens de programação, arquiteturas de integração de sistemas – SOA –, protocolos de comunicação – SOAP –, entre outros.

3.3 FORMALIZANDO A ARQUITETURA DE CONTROLE

A proposta para a implementação física do sistema de controle e supervisão, adota a Arquitetura de Controle Supervisório definida em (Queiroz e Cury, 2002), no qual é constituída por três níveis conforme ilustrado na Figura 3.4.

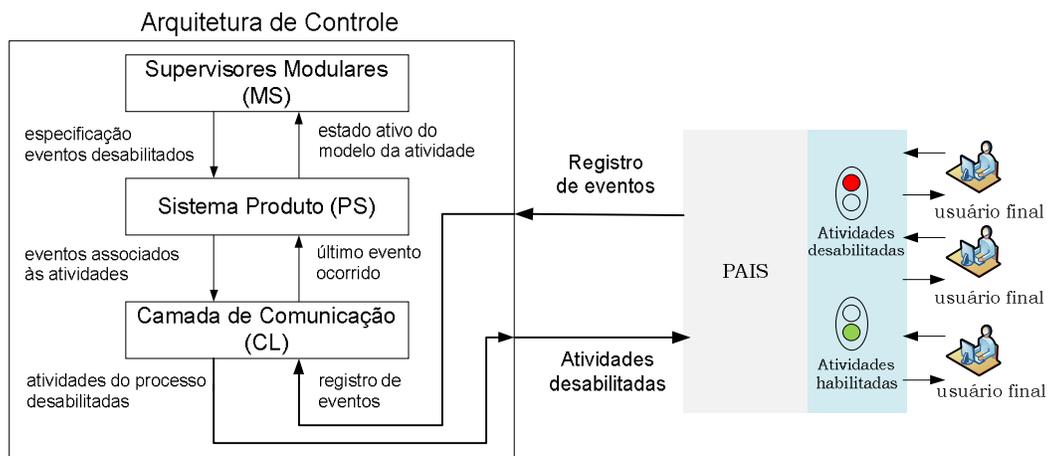


Figura 3.4 - Arquitetura do Sistema de Controle e Supervisão

A arquitetura de controle está estruturada da seguinte forma:

- a) Supervisores Modulares – (MS): neste nível são implementados os supervisores obtidos na etapa de síntese; tanto pode ser utilizado um único

supervisor monolítico quanto um conjunto de supervisores modulares locais;

- b) Sistema Produto – (PS): neste nível são implementados os modelos das atividades identificadas na etapa de modelagem;
- c) Camada de Comunicação – (CL): este nível é responsável pelo gerenciamento da comunicação entre o sistema supervisionado – PAIS – e o sistema de controle e supervisão –SCS.

A comunicação entre os diferentes níveis do SCS é representada na Figura 3.4. Conforme descrito anteriormente, cada atividade é constituída por diversos eventos. Os usuários do sistema são os responsáveis pelo disparo de tais eventos. A cada ocorrência destes eventos no PAIS a Camada de Comunicação receberá o conteúdo do registro de eventos. Tal registro possui várias informações a respeito da execução das atividades, porém somente a informação a respeito da ocorrência dos eventos é importante para o SCS. Desta forma, a camada de comunicação irá tratar o registro de eventos, a fim de identificar qual o último evento ocorrido. A ocorrência deste evento é sinalizada para o nível Sistema Produto para que ocorra a atualização do estado ativo do modelo que representa a atividade associada a este evento e para que a ocorrência do mesmo seja efetivamente sinalizada ao nível Supervisores Modulares. O estado ativo de cada um dos supervisores deve ser devidamente atualizado em função da ocorrência de tal evento. Também deve ocorrer a atualização da ação de controle dos supervisores, ou seja, do conjunto de eventos associados às atividades que devem ser desabilitados. Esta informação é encaminhada diretamente para a Camada de Comunicação para que seja devidamente tratada e encaminhada ao PAIS.

3.4 CONCLUSÃO DO CAPÍTULO

Neste capítulo foi apresentada a visão geral do funcionamento do SCS, além da abordagem de controle e supervisão a ser utilizada na presente pesquisa. Para elaborar um sistema de controle e supervisão, identificou-se a necessidade de realização de quatro etapas: Modelagem, Síntese, Implementação, Validação e

Testes. Estas etapas fazem parte de um ciclo contínuo e tem como objetivo sempre manter o SCS atualizado. Esta característica vem de encontro com as práticas recomendadas pelo BPM, que prega a melhoria contínua dos processos.

Após a apresentação do ciclo, formalizou-se a estrutura de controle e supervisão a ser aplicada no SCS.

4 PROJETO DO SCS

Neste capítulo, inicialmente é apresentada uma visão geral sobre o ProM Framework, plataforma utilizada para o desenvolvimento do SCS. Também é apresentada a especificação da estrutura projetada para o desenvolvimento do SCS. Nesta especificação são demonstrados os modelos que foram criados para a elaboração e o desenvolvimento do SCS bem como a arquitetura utilizada para a sua implementação. Para modelar e apresentar as principais características do SCS foi utilizada a linguagem UML – Unified Modeling Language. O desenvolvimento do SCS foi realizado utilizando os recursos da plataforma Java Enterprise Edition – JEE e Arquitetura Orientada a Serviços – SOA.

4.1 PROM FRAMEWORK

O ProM Framework é uma plataforma que agrupa uma série de ferramentas (plug-ins) que permitem a utilização da mineração de processos para a análise dos processos de negócio. Por meio da mineração de processos é possível extrair informação não-trivial e útil dos chamados "registro de eventos", gerados durante a execução das atividades dos processos (AALST et al. 2007).

A mineração de processos abrange técnicas que permitem aprofundar a análise dos dados dos processos e identificar situações que podem auxiliar no processo de melhoria contínua. Uma das técnicas mais populares da mineração de processos é a técnica de descoberta do fluxo do processo. Por meio desta técnica é possível realizar a construção de um modelo de processo (por exemplo, uma rede de Petri - RdP), que descreve as dependências de causalidade entre as atividades. No entanto, utilizar esta técnica na prática, sem o apoio de uma ferramenta, é uma tarefa muito complexa. Neste cenário, o ProM Framework fornece uma plataforma robusta que analisa os registros de eventos do processo a ser avaliado e constrói o fluxo do processo em RdP com base nas técnicas de mineração de processos.

O ProM Framework no entanto, não se limita apenas à técnica de descoberta do fluxo do processo. Ele também permite a descoberta de outras perspectivas (por

exemplo, dados e recursos) e suporta técnicas afins, tais como a verificação de conformidade, o modelo de extensão, a transformação do modelo, verificação, etc (AALST et al. 2007). Estas características tornam o ProM uma ferramenta versátil para a análise de processo que não se restringe a análise do modelo, mas também inclui a análise baseada em logs de eventos. A figura 4.1 ilustra uma visão geral das técnicas de mineração de processos disponíveis no ProM.

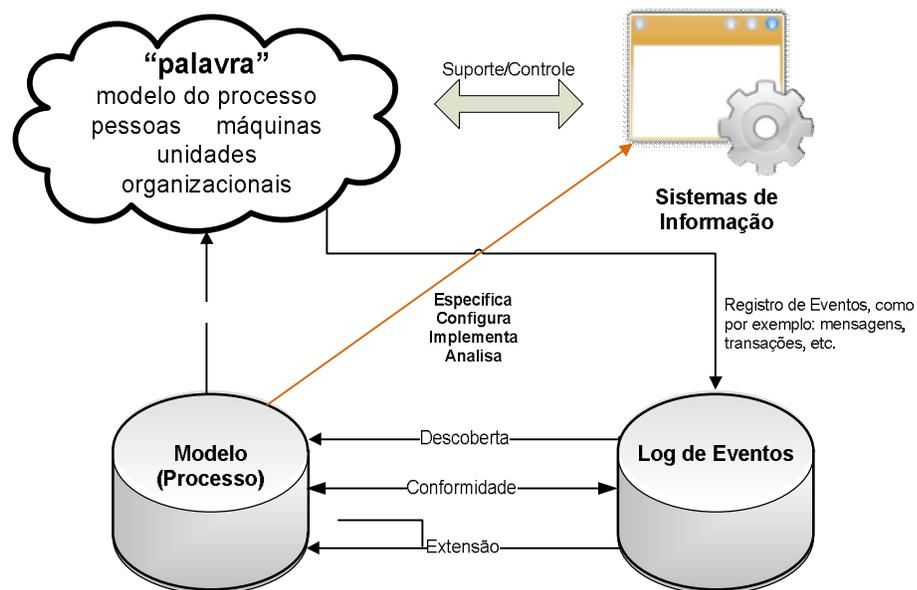


Figura 4.1–Visão geral das técnicas de mineração no ProM
Fonte: Aalst et al. (2007)

As técnicas de mineração de processos que foram implementadas no ProM, por meio das ferramentas disponíveis, têm como principal requisito a análise dos dados da realidade observada dos processos de negócio. Para isso, o ProM utiliza os modelos (por exemplo, RdP), e estes modelos podem ser utilizados como segue: (1) são descobertos a partir de registro de eventos, (2) são usados para refletir sobre a realidade observada (verificação de conformidade) ou (3) são estendidos com base em informações extraídas de registros (AALST et al. 2007).

A versão 1.1 do ProM Framework foi apresentada na *Petri Net Conference*, em 2005. Desde a apresentação até os dias atuais, o ProM foi consideravelmente estendido e, atualmente, dezenas de pesquisadores estão desenvolvendo plug-ins para o framework. O ProM é uma ferramenta livre e utiliza uma arquitetura que permite a inclusão de novas ferramentas, por exemplo, onde as pessoas podem

adicionar novas técnicas de mineração de processo por meio da adição de plug-ins sem gastar esforços no sentido do carregamento e filtragem de registros de eventos e na visualização dos modelos resultantes, pois estes recursos já são oferecidos pela plataforma do ProM Framework. Esta foi uma das motivações para que o ProM fosse utilizado como plataforma para o desenvolvimento do SCS, visto que os recursos para o tratamento dos registros de eventos poderiam ser reaproveitados, reduzindo assim o custo do desenvolvimento do SCS.

4.2 ESPECIFICAÇÃO E IMPLEMENTAÇÃO DO SCS NO PROM

De acordo com o conceito do SCS discutido no capítulo anterior, este serviço tem como objetivo monitorar a execução das atividades de um processo de negócio implementado em um PAIS. No entanto, o SCS é um serviço independente do PAIS. O SCS é um serviço externo que presta suporte ao PAIS. A principal função do SCS é verificar como as atividades estão sendo realizadas e auxiliar os usuários finais na execução das demais atividades de acordo com a ação de controle preestabelecidos.

A fim de garantir a independência do SCS e a possibilidade de uso futuro em outros PAIS, o SCS foi implementado como um plug-in no ProM Framework. Como já citado no tópico anterior, o ProM é um *framework* que disponibiliza uma série de plug-ins para o suporte das práticas de BPM. Devido a esta característica, considerou-se como uma alternativa interessante para evolução e continuidade do SCS, a implementação deste como um serviço disponível no ProM.

O SCS foi implementado no ProM como um provedor do *Operational Support* (OS). O OS é um *plug-in* que permite a interação entre os serviços disponibilizados no ProM e o meio externo. O OS fornece uma interface TCP/IP para a comunicação dos serviços no ProM com o PAIS externo. Desta forma, os PAISs podem solicitar informações do OS em qualquer momento durante a execução de uma instância do processo. Quando uma solicitação é feita ao OS, esta solicitação deve conter (1) o histórico da instância do processo na forma da lista de atividades que foram executadas até o momento da solicitação, e (2) a lista de atividades que estão ativas no momento da solicitação. O papel do OS então é atuar entre os serviços

implementados no ProM e o meio externo. Esta arquitetura é necessária, pois o ProM disponibiliza vários serviços para o gerenciamento de processos, sendo o SCS apenas um destes serviços que poderão ser utilizados. A visão geral da aplicação do SCS como um provedor do OS no ProM é apresentada na figura 4.2.

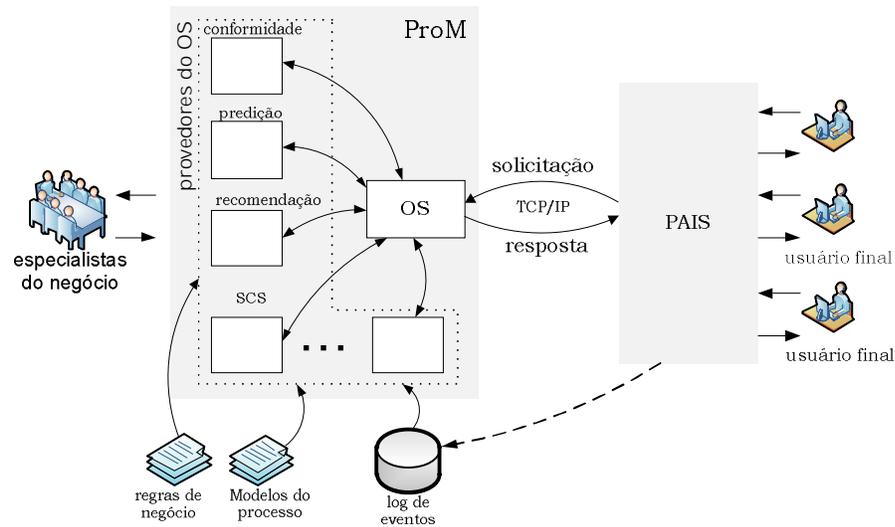


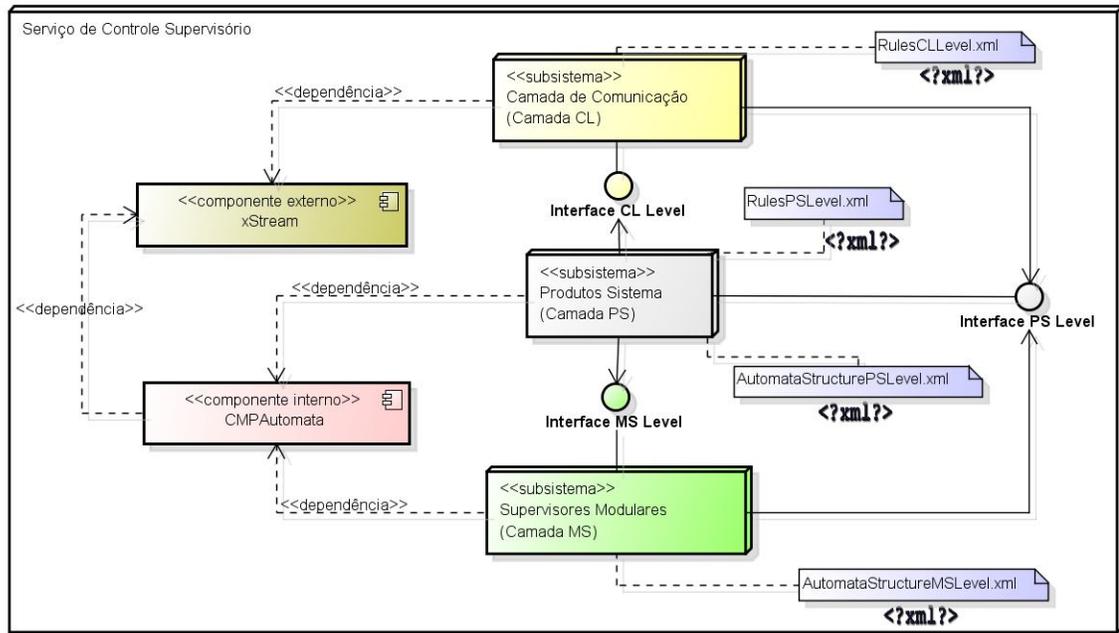
Figura 4.2 - Visão Geral dos provedores do OS no ProM

Conforme demonstrado na figura 4.2, o OS pode ter ligações com um número arbitrário de provedores (serviços) que implementam vários tipos de apoio operacional. Quando o OS recebe uma solicitação do PAIS, este encaminha a solicitação presente a todos os provedores conectados e envia as respostas para o PAIS. Isto permite vários tipos de apoio operacional ao mesmo tempo. Por exemplo, o provedor de controle e supervisão, o SCS, responde enviando o guia com a lista de atividades desabilitadas, com base no acompanhamento das regras de negócio e perfil do usuário.

4.2.1 ARQUITETURA DO SCS

A arquitetura do SCS é ilustrada na figura 4.3. A arquitetura do SCS foi implementada seguindo a formalização da arquitetura de controle – SCSA – especificada no tópico 3.3 do capítulo 3. Portanto, o SCS utiliza em sua estrutura

física, os três níveis com base na abordagem de controle modular: Camada de Comunicação (nível CL), Sistema Produto (nível PS) e Supervisores Modulares (nível MS). O SCS também utiliza em sua estrutura física dois componentes: um desenvolvido especificamente para o SCS, o componente interno chamado CMPAutomata e o componente externo chamado xStream.



powered by astah

Figura 4.3 - Arquitetura do SCS

Como ilustrado na figura 4.3, o SCS utiliza informações que ficam armazenadas como parâmetros em arquivos XML. As informações acerca das regras de controle, especificação das regras dos autômatos, são armazenadas em dois arquivos: `AutomataStructurePSLevel.xml` e `AutomataStructureMSLevel.xml`. O primeiro arquivo, possui as especificações do nível PS, que implementa os supervisores locais, e o segundo arquivo possui as especificações do nível MS, os supervisores modulares. Os arquivos XML são lidos pelo SCS por meio do componente externo `xStream`. Este componente é um componente padrão que permite a manipulação de arquivos XML, tanto para leitura quanto para escrita. Os arquivos XML: `RulesCLLevel.xml` e `RulesPSLevel.xml`, por sua vez, possuem algumas regras acerca das atividades dos processos. Nestes arquivos são indicados quais e quantas são as atividades dos processos e os possíveis eventos

que podem ocorrer na execução de cada atividade. O arquivo: RulesCLLevel.xml também possui a especificação dos perfis de usuário do processo que serão utilizados pelo SCS. Para a presente pesquisa foram utilizados dois perfis: (1) operador do processo e (2) analista do processo. O operador do processo será utilizado para os usuários que devem executar as atividades de acordo com as regras de negócio predefinidas. O analista de processo, por sua vez, será utilizado pelos usuários que possuem conhecimento acerca do processo de negócio e, portanto, poderão visualizar o guia com as regras de negócio aplicadas para a execução das atividades do processo, porém não terão nenhum tipo de restrição caso desejem executar uma atividade que esteja indicada como desabilitada pelo guia do SCS.

A Figura 4.4 ilustra o diagrama de classe UML das classes mais importantes do SCS. A classe principal é *SupervisorControlProvider* que é responsável por (1) receber o log dos eventos contendo a lista das atividades que foram realizadas até o momento na instância do processo em análise e (2) enviar essa informação para os níveis de controle do SCS. O log de eventos é recebido pelo OS em formato de requisição e é encaminhado para os provedores para análise. A classe *SupervisoryControlProvider* recebe a requisição e faz a interface entre o OS e os níveis do SCS.

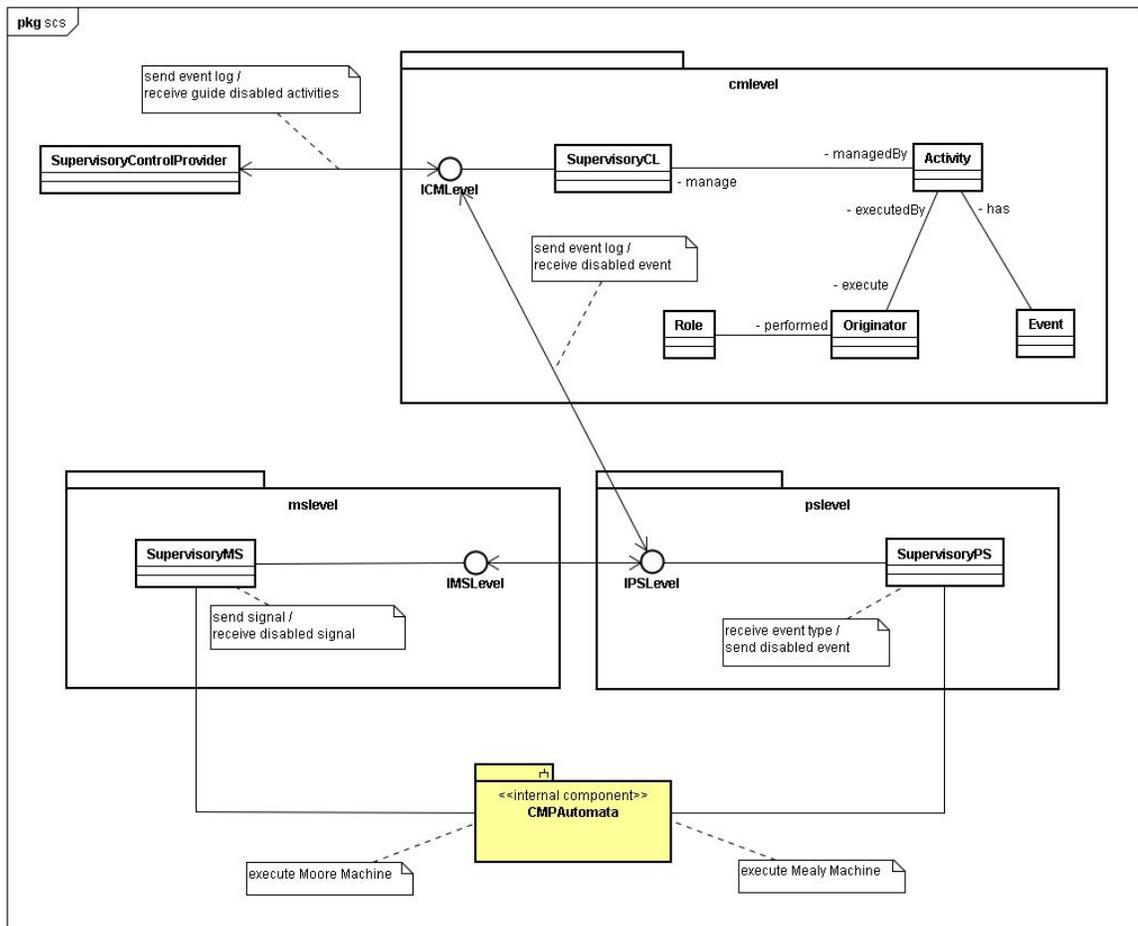


Figura 4.4 - Diagrama de Classes do SCS

Para habilitar a interface entre o OS e o SCS, a classe *SupervisoryControlProvider* estende a classe abstrata *AbstractProvider*. Esta classe é mantida pelo OS e tem como objetivo possibilitar a interface entre o OS e o provedor de serviço. Ao estender a classe *AbstractProvider* é necessário também implementar o método:

{ ... }

Por meio da implementação deste método, é possível a criação de uma resposta a cada requisição (log de eventos) recebida. A figura C.1 do apêndice C ilustra a implementação da classe *SupervisoryControlProvider* estendendo a classe *AbstractProvider*. A figura 4.5 ilustra o diagrama com as principais classes do OS.

Quando a requisição é recebida pelo *SupervisoryControlProvider*, o log de evento é enviado ao nível CL para iniciar a execução do SCS. No final da execução, a classe *SupervisorControlProvider* receberá atividades do CL que devem ser desabilitadas, para que sejam retornadas ao OS no formato de resposta adequado. O formato de resposta utilizado é implementado por meio da classe: *ProviderResponse* do OS.

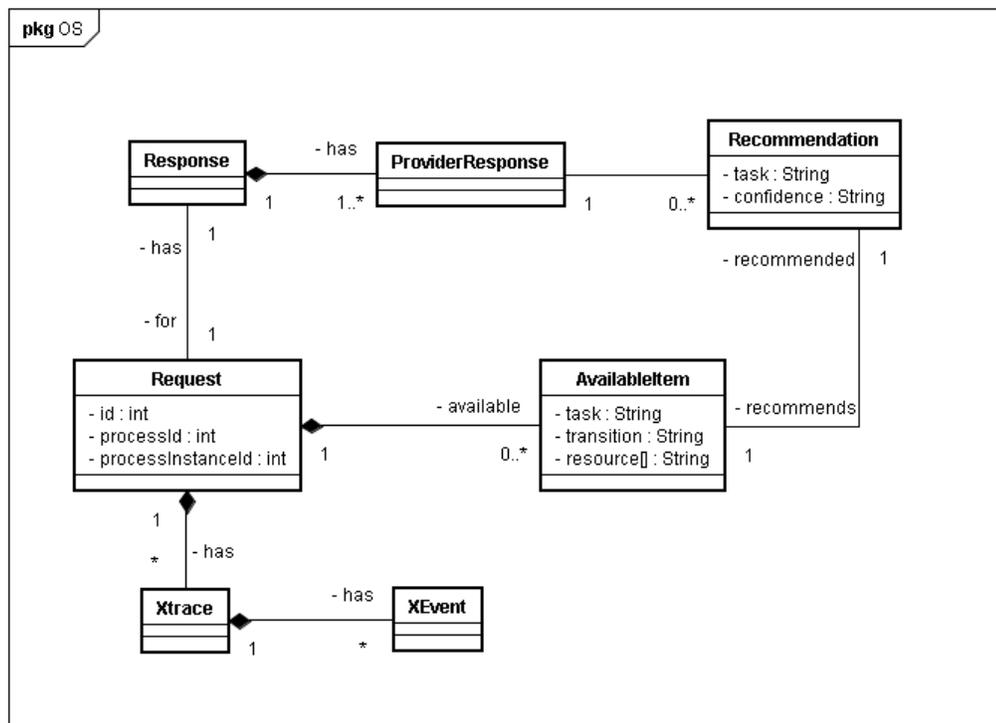


Figura 4.5 - Diagrama de Classes OS

A comunicação entre PAIS, o OS e o SCS é detalhado na figura 4.6. A comunicação começa quando o PAIS envia uma solicitação de serviço ao OS. Esta solicitação inclui as seguintes informações: identificação da solicitação, identificação do processo, identificação de instância do processo que está em execução, além dos dados do registro de eventos e perfil do usuário. Ao receber a solicitação, o OS encaminha para os provedores de serviço. No caso do SCS, a solicitação é recebida primeiro pela classe *SupervisoryControlProvider*. Essa classe receberá a solicitação com todas as informações do registro de eventos e enviará ao nível CL. Neste momento, tem início a execução do SCS e de seus níveis. A seqüência de execução e o compartilhamento das informações entre os níveis são ilustrados nas figuras 4.6 e 4.7.

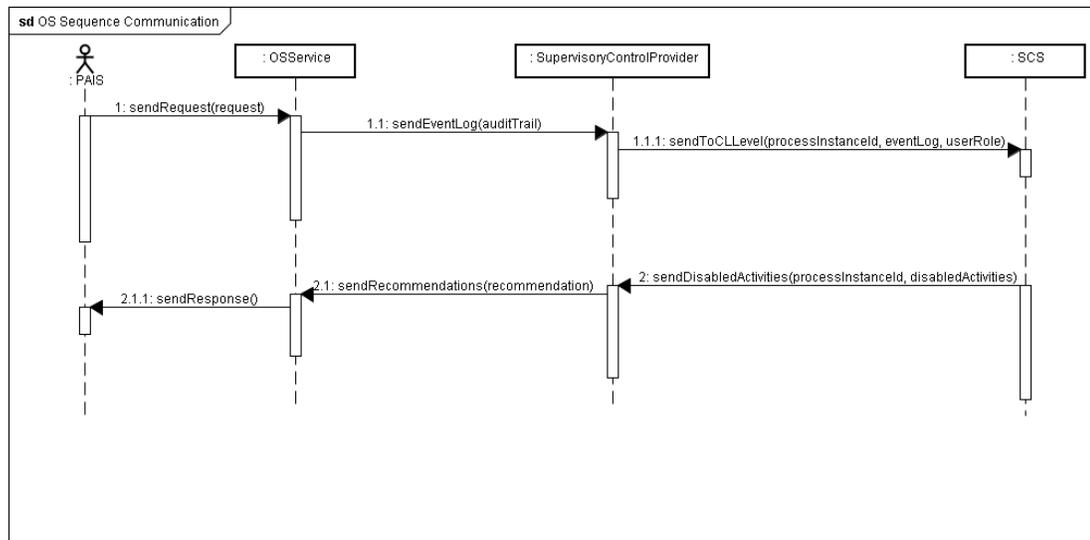


Figura 4.6 - Comunicação entre o PAIS, OS e SCS

Na figura 4.7 é ilustrada a comunicação entre os níveis do SCS, quando ocorre o recebimento de uma solicitação do PAIS. A primeira classe do SCS a receber a informação a respeito da solicitação do PAIS é a classe: *SupervisoryControlProvider*. Como esta classe implementa a classe *AbstractProvider* do OS, assim que o OS recebe uma solicitação, a solicitação já é compartilhada entre todos os provedores de serviço que implementam esta classe e, entre eles está o SCS. A classe *SupervisoryControlProvider* possui um método que lê os logs de eventos recebidos do PAIS por meio do OS. O OS recebe os logs de eventos no formato da classe: *Request*. Esta classe possui relação com a classe: *XTrace*, que por sua vez possui relação com a classe *XEvent*, como demonstra o diagrama de classes na figura 4.5. A classe *XTrace* armazena os dados acerca dos eventos que ocorreram durante a execução do processo e a classe *XEvent* especifica os detalhes de cada evento ocorrido.

A classe *XTrace*, no entanto, armazena todo o histórico dos logs de eventos que ocorreram durante a execução do processo. Para o SCS, porém, é necessária apenas a informação do último evento que foi disparado pelo PAIS, pois este é considerado o evento atual do fluxo do processo. A classe *SupervisoryControlProvider* possui o método: *readEventLog* que tem como objetivo extrair a informação do último evento disparado pelo PAIS. Após obter esta informação a classe *SupervisoryControlProvider* encaminha a informação do último evento para a interface do CL, implementada por meio da classe: *SupervisoryCL*. O

CL por sua vez irá extrair informações acerca do tipo do evento e encaminhar para a interface do PS, implementada por meio da classe: *SupervisoryPS*. Ao receber o último evento ocorrido o PS irá executar os supervisores locais, identificar qual é o estado ativo do modelo que representa a atividade do evento ocorrido e encaminhar para a interface do MS, implementada pela classe: *SupervisoryMS*. A execução dos supervisores locais por meio do PS, somente ocorre se o tipo de evento é diferente de vazio. Um tipo de evento com valor vazio pode ser gerado, por exemplo, quando um PAIS é iniciado e sua comunicação com o SCS é estabelecida. Neste caso, nenhum evento de processo ocorreu, o que ocorreu foi apenas uma configuração dos sistemas e seus mecanismos de integração. Então, quando o PS receber um tipo de evento vazio, este encaminha uma informação com valor nulo direto para a interface do MS. O MS verifica qual o valor da informação recebida e no caso de ser um valor nulo, retorna as regras de controle com base na última execução do processo em execução. Quando o valor da informação for diferente de nulo, o MS irá executar os supervisores modulares, identificar as regras de controle e retornar para o PS.

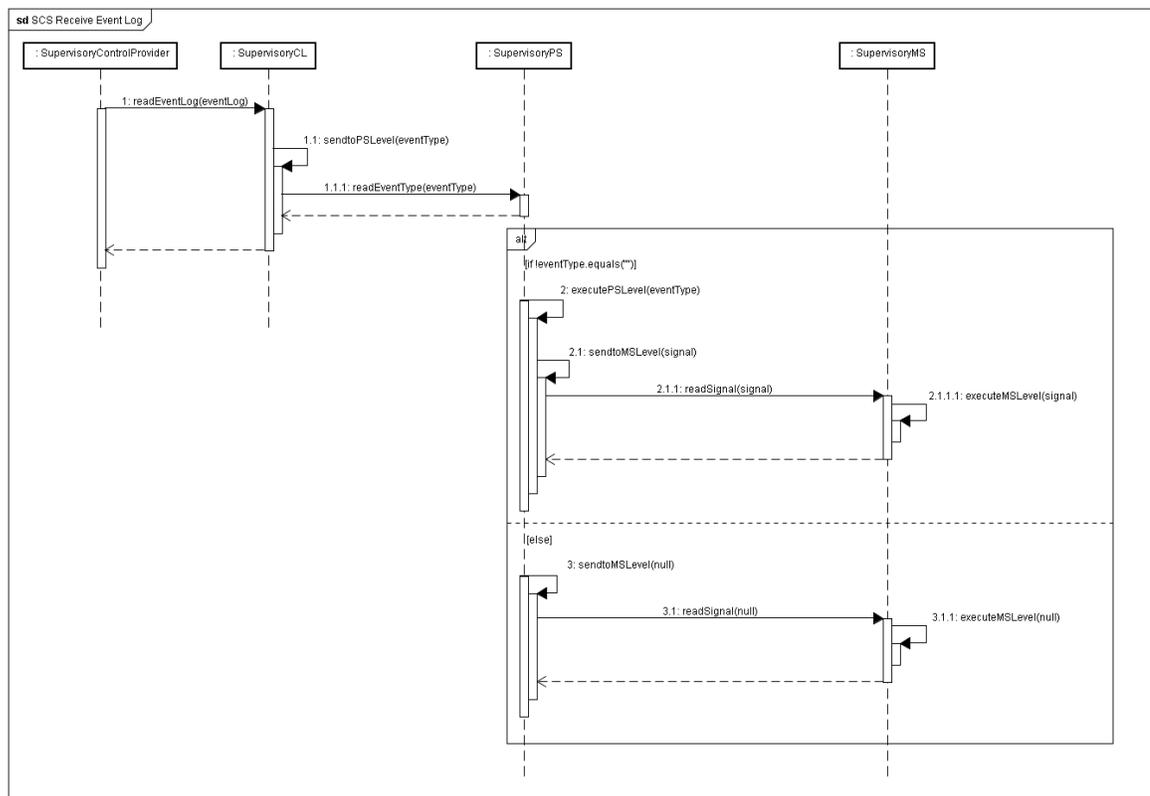


Figura 4.7 - Comunicação entre os níveis do SCS – Recebimento de Solicitação

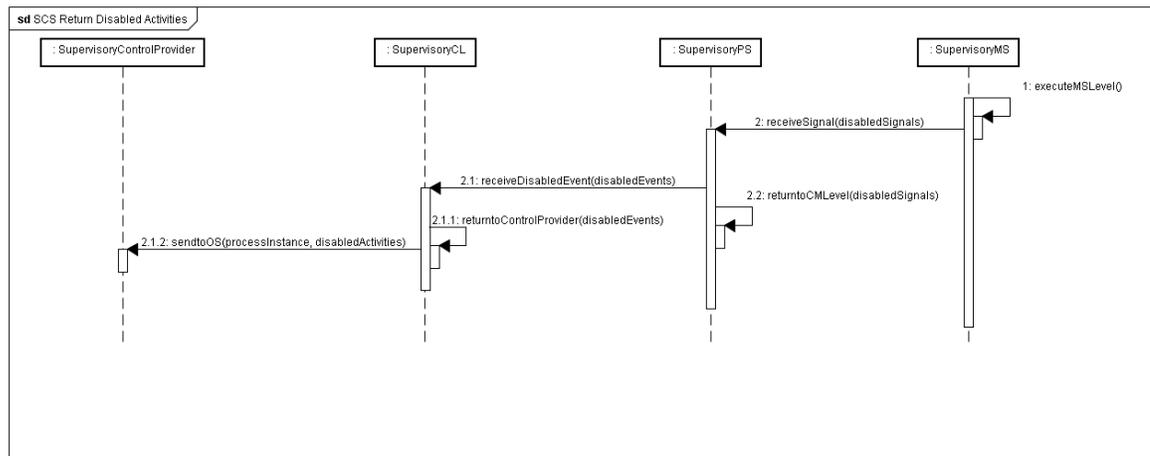


Figura 4.8 - Comunicação entre os níveis do SCS - Retorno da Execução

A figura 4.8 ilustra a comunicação entre os níveis do SCS, no retorno da execução do SCS. O retorno da execução do SCS ocorre após a execução do MS. Como já mencionado, o MS verifica o valor da informação recebida e encaminha os eventos a serem desabilitados para a interface do PS. O PS identifica os eventos que devem ficar desabilitados, com base no retorno do MS e encaminha esta informação para a interface do CL. O CL recebe os eventos desabilitados, localiza nos parâmetros, o arquivo *RulesCSLevel.xml*, quais atividades estão relacionadas aos eventos desabilitados, e monta o guia de atividades para retornar para o PAIS. Antes de encaminhar o guia para o PAIS, o CL também verifica nos parâmetros, qual o papel do usuário no processo e, caso este possua perfil do operador, atribui ao guia a informação de restrição das atividades. Esta informação será utilizada pelo PAIS e pelo próprio SCS para restringir a execução das atividades indicadas como desabilitadas.

Quando o SCS finaliza a execução o nível CL retorna para a classe *SupervisoryControlProvider* a identificação da instância do processo e o guia das atividades. Este guia de atividades é construído com base nos logs de eventos recebidos no início da execução. É necessário retornar a identificação da instância do processo para a classe *SupervisoryControlProvider*, pois o SCS permite a execução de mais de uma instância de processo. Portanto, para que seja mantida a integridade das informações de cada instância, o SCS foi implementado utilizando recursos de transação. Ao final, o nível CL retornará as informações para o OS que transmitirá para o meio externo, o PAIS que fez a solicitação.

4.2.2 CMPAutomata

Nesta seção serão descritas as principais características sobre o componente interno CMPAutomata. Este componente é responsável pela verificação da instância do processo em execução e pela execução dos autômatos dos supervisores locais e autômatos dos supervisores modulares. A figura 4.9 ilustra o diagrama de classe UML das classes mais importantes do componente CMPAutomata. A principal classe do componente é a *Supervisory*. A instância do processo – case – foi definida como *token* na estrutura do componente, pois na linguagem Java o termo *case* é uma palavra reservada.

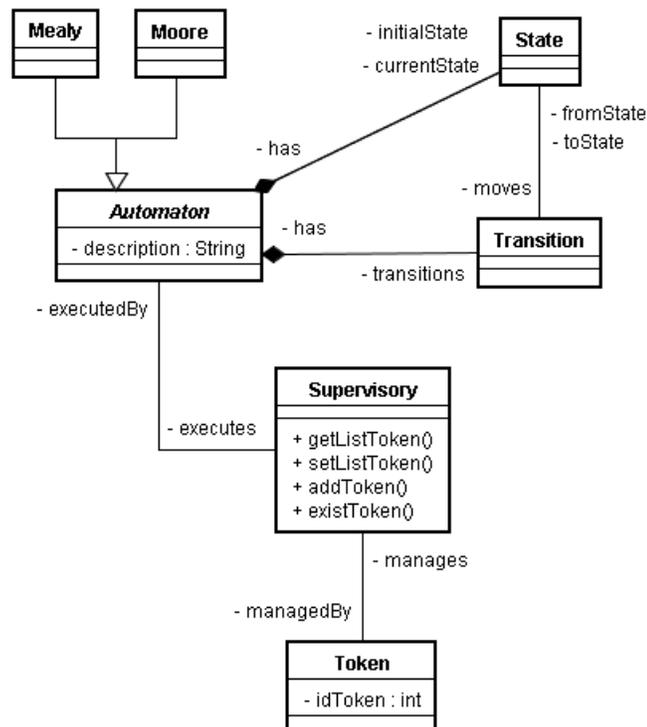


Figura 4.9 - Diagrama de Classes do componente CMPAutomata

Na figura 4.9, a classe *Token* representa a estrutura de um objeto que implementa a instância de um processo. A cada instância de processo recebida, será criado um novo objeto do tipo *Token*. Esta estrutura permite que o SCS analise

as informações de mais de uma instância de processo por vez, conforme os recursos de transação disponíveis na plataforma Java. A classe *Automaton* é a classe responsável pela aplicação das regras de controle armazenadas nos arquivos XML: *AutomataStructurePSLevel.xml* e *AutomataStructureMSLevel.xml*.

As classes *Moore* e *Mealy* são especializações da classe *Automaton*, portanto, herdam todas as características da classe *Automaton*. A classe *Moore* implementa o conceito da Máquina de Moore e será executada por meio do nível MS. A classe *Mealy*, por sua vez, implementa o conceito da Máquina de Mealy e é executada por meio do nível PS. As transições dos autômatos serão controlados pela classe *Transition* e os estados pela classe *State*. Para esta arquitetura, um autômato possui dois estados: o estado inicial e o estado atual. Estes estados são representados pelos atributos “fromState” e “toState”.

4.3 CONCLUSÃO DO CAPÍTULO

Neste capítulo foi apresentada uma visão geral acerca do ProM Framework, plataforma utilizada para o desenvolvimento do SCS. Também foi apresentada a especificação da arquitetura elaborada para o desenvolvimento do SCS. A arquitetura foi desenvolvida com base na arquitetura SCSA detalhada no capítulo 3. O SCS foi implementado como um provedor de serviços do ProM. Este tipo de serviço responde as solicitações do OS que é o plug-in do ProM responsável pelo recebimento e controle das solicitações do PAIS no meio externo. O SCS foi desenvolvido na plataforma Java Enterprise Edition, com recursos de SOA para permitir a integração com os PAISs. Esta plataforma foi selecionada para o desenvolvimento, pois é a plataforma padrão para o desenvolvimento de plug-ins e serviços no ProM.

5 CONFIGURAÇÃO E EXECUÇÃO DO SCS

Neste capítulo são demonstrados o exemplo de modelo de processo selecionado e suas restrições de controle para a condução do experimento, além dos procedimentos e recursos utilizados para a configuração e execução do processo de negócio no PAIS supervisionado pelo SCS. Inicialmente apresentam-se o modelo do processo e as regras de negócio utilizadas para a aplicação do SCS. Também são apresentados os recursos de tecnologia da informação implementados para possibilitar a integração do SCS em dois PAIS: o Bizagi e o YAWL, além da arquitetura completa desenvolvida para a implementação do SCS. Na seqüência são apresentadas as configurações necessárias para iniciar os testes com a execução do SGA nos dois PAISs sob a supervisão do SCS. Estas configurações são pré-requisito para a utilização do SCS como suporte para a execução dos processos em um PAIS. As configurações que devem ser realizadas nos PAISs são apresentadas resumidamente neste capítulo e, detalhadamente nos apêndices. Por fim, são apresentados o plano de testes e os casos de testes elaborados para a realização do experimento a fim de validar a abordagem proposta no presente trabalho.

5.1 SISTEMA DE GERENCIAMENTO DE ANOMALIAS – SGA

Com o objetivo de exemplificar a aplicação do SCS em um processo de negócio real, utilizou-se o Sistema de Gerenciamento de Anomalias (SGA). O SGA tem como fundamento a padronização da resolução de problemas relacionados a não-conformidades encontradas nos processos de negócio de uma indústria. Em linhas gerais, o SGA engloba o planejamento (identificação das causas da anomalia), o plano de ação (registro das atividades), a verificação (análise da melhoria implementada) e a padronização e efetividade das ações implementadas e diagnosticadas. Este último possui o objetivo principal de garantir que as ações realizadas repercutiram em um resultado satisfatório para o tratamento da anomalia encontrada.

Em alinhamento e conformidade com os requisitos da ABNT ISO 9001 (2000), a proposta de um SGA na empresa nasceu da necessidade de registrar e evidenciar a melhoria contínua e a correção de não-conformidades ocorridas dentro do fluxo produtivo. O SGA engloba as diversas áreas e processos da companhia e possui como principal objetivo compilar informações necessárias para a resolução de problemas originados das auditorias em produtos acabados e nos seus respectivos processos de manufatura.

Tais problemas, identificados no processo como anomalias, podem ter origens diversas e, portanto, devem ser devidamente identificados e gerenciados, com o objetivo principal de eliminar a causa “raiz” do problema. Visando um controle eficaz, a empresa optou por um SGA informatizado de suporte e controle, focado no gerenciamento da informação. A informação a ser gerenciada, neste caso, é toda aquela que se refere à origem e identificação da anomalia, bem como a sua forma de tratamento para correção e posterior prevenção. Além destas funções, o SGA possibilita o fluxo das informações e o compartilhamento das responsabilidades. O sistema de informação que suporta e gerencia as informações do SGA possui as características de um sistema de gerenciamento de fluxo de trabalho – *Workflow*.

O presente trabalho objetiva utilizar o modelo do fluxo do processo do SGA com o suporte do SCS a fim de demonstrar a aplicabilidade do sistema de controle e supervisão como uma forma para aumentar a capacidade de gerir os processos de negócio das organizações. O modelo do processo do SGA é apresentado nas figuras A.1 e B.1 dos apêndices A e B, sendo a figura A.1 a ilustração do modelo do processo elaborado na notação BPMN e a figura B.1 na notação YAWL.

Analisando o modelo, o processo é iniciado por meio do evento: Ocorrência de Anomalia que inicializa a atividade: Identificação da Anomalia. Após a identificação e registro da anomalia, são validadas as informações que foram preenchidas no registro. Esta atividade é realizada pela liderança da área destinatária do registro. Se o líder encontrar alguma inconsistência no registro, o registro é retornado ao emissor da anomalia para correção das informações, esta situação é caracterizada pela ocorrência do evento: Registro incorreto. Se o registro da anomalia estiver adequado, o evento: Anomalia Registrada será disparado e o registro é encaminhado para a atividade: Elaboração do Plano de Ação.

Nesta atividade, é iniciado o diagnóstico das causas geradoras da anomalia bem como o preenchimento do plano de ação para eliminação destas causas. Após

a realização desta atividade, serão disponibilizadas ao usuário final cinco atividades para serem realizadas. As atividades que serão realizadas são: Definição dos Por quês, Identificação das Causas, Definição das Ações Corretivas, Definição das Ações Preventivas e Definição de Proposta de Eficácia. A ordem e priorização na realização destas atividades são feitas pelo próprio usuário final, conforme os critérios estabelecidos pelo mesmo. Nesta etapa do processo, a utilização do SCS para guiar o usuário final, permitirá monitorar e restringir a execução das atividades do processo em função das restrições impostas para garantir o alcance dos objetivos acerca do tratamento da anomalia que está sendo realizado.

Após a realização destas atividades, a proposta para o plano de ação é gerada e encaminhada para a liderança do setor. A liderança valida a proposta do plano de ação e autoriza a sua execução. Porém, caso exista alguma inconsistência na proposta do plano de ação, o líder devolve a proposta para ser alterada; esta situação é caracterizada pela ocorrência do evento: Plano de Ação Insuficiente. A atividade de validação pela liderança acontece novamente após a execução do plano de ação. Nesse caso, a liderança avalia os resultados alcançados após a execução do plano de ação e decide se as ações foram efetivas ou não para resolver todos os problemas encontrados. Caso os resultados identificados não foram efetivos, o relatório de anomalia retorna para a área de responsabilidade para revisão do plano de ação. Caso contrário, a data de término das ações implementadas é registrada, as atividades são finalizadas e o registro da anomalia é armazenado para fins históricos.

As regras de negócio – restrições de controle – das atividades: Identificação das Causas, Definição das Ações Corretivas, Definição das Ações Preventivas, Definição dos Por quês e Definição de Proposta de Eficácia são apresentadas na tabela 5.1.

Tabela 5.1–Relação entre as atividades do SGA e restrições de controle

ID da Atividade	Descrição da Atividade	Restrições
B	Definição das Ações Corretivas	Estas atividades, utilizam o mesmo aplicativo de banco de dados e caso a atividade «B – Definição das Ações Corretivas» seja executada e, logo na seqüência, a atividade «C – Definição das Ações Preventivas» seja executada, a duração combinada das duas atividades é menor.
C	Definição das Ações Preventivas	

		dos dados fornecidos por «B – Definição das Ações Corretivas», eliminando desta forma a necessidade de entrada destes dados.
A	Identificação das Causas	Estas atividades, durante a sua execução, podem falhar inesperadamente.
E	Definição de Proposta de Eficácia	No caso de ambas as atividades falharem simultaneamente, a reparação da tarefa «A – Identificação das Causas» terá prioridade sobre a tarefa «E – Definição de Proposta de Eficácia».
B	Definição das Ações Corretivas	Por razões de segurança, estas atividades não devem ser executadas ao mesmo tempo, pois é necessária uma exclusão mútua entre elas.
D	Definição dos Por quês	

De acordo com as restrições apresentadas na tabela 5.1, o SCS poderá então orientar os usuários informando quais atividades estão desabilitadas, a fim de cumprir as especificações de controle e, no caso do usuário possuir um perfil de operador, o SCS pode também restringir a execução destas atividades para evitar a violação das regras de negócio definidas.

Considerando o ciclo de desenvolvimento de controle e supervisão, a primeira etapa para o desenvolvimento deste tipo de sistema é a modelagem. Nesta etapa, cada atividade foi modelada como um autômato de dois estados: (1) um estado inicial, que significa que a atividade não está sendo executada, pois a instância do processo ainda não chegou e (2) outro estado, que significa o início de processamento da instância do processo. No caso das atividades *A – Identificação das Causas* e *E – Definição da Proposta de Eficácia*, também é necessário representar um terceiro estado, (3) estado que representa a ocorrência da falha. Os autômatos que representam as atividades são chamados: G_A , G_B , G_C , G_D e G_E . A figura 5.1 ilustra os autômatos. Os estados “0” representam o estado inicial, onde a instância do processo ainda não foi recebida e, os estados “1” representam a execução da instância do processo, quando foi recebido um registro de eventos com os eventos ocorridos na execução da atividade. Para os autômatos G_A e G_E os estados “2” representam a falha e a suspensão na execução da atividade. Os eventos que ocorrem na execução das atividades são representados pelas palavras:

- a) *start*: início da execução da atividade;
- b) *complete*: finalização da execução da atividade;
- c) *suspend*: atividade suspensa devido a alguma ocorrência de falha;
- d) *repair*: reparação da atividade.

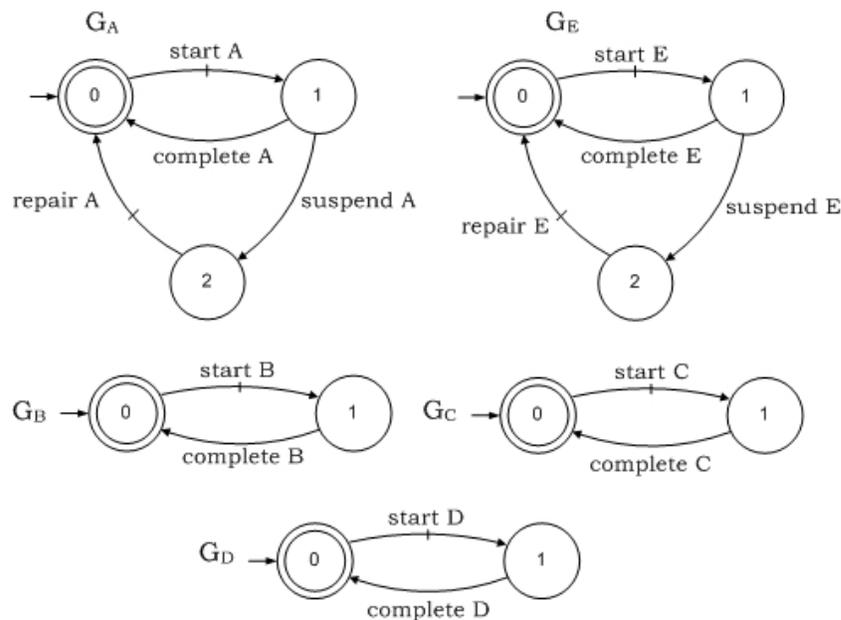


Figura 5.1 - Representação dos Autômatos das Atividades

O evento relacionado ao início da atividade, *start*, é considerado um evento controlável e, o evento relacionado a finalização da atividade, *complete*, é considerado um evento não-controlável. Desta forma, quando uma atividade é iniciada o supervisor que controla essa atividade não pode mais impedir a sua finalização. Além disso, o evento relacionado à ocorrência de falhas na execução das atividades, *suspend*, é considerado um evento não-controlável, pois este ocorre de forma imprevista. Por fim, o evento relacionado a reparação da atividade, *repair*, é considerado um evento controlável devido a sua natureza de permitir ou não a reparação da atividade. Para representar os eventos controláveis, na figura 5.1, os arcos que representam os eventos controláveis possuem uma linha ao centro do arco.

A figura 5.1 ilustra o modelo de autômatos com a representação de todos os eventos que podem ocorrer no PAIS supervisionado,. Neste caso, esses autômatos representam a semântica das atividades envolvidas no processo, sem quaisquer restrições de controle. Portanto, é necessário incluir algum controle durante a execução das atividades para atingir as especificações das regras de controle. A inserção deste controle pode ser feita evitando algumas seqüências de eventos. Para calcular a lógica de controle que evita a ocorrência de seqüências indesejadas, é necessário formular as especificações do problema, através de autômatos, de

forma a representar o comportamento desejado ou aquele que se deseja impedir. Esta formulação das especificações representa a segunda etapa do ciclo de desenvolvimento do controle e supervisão.

As especificações foram definidas considerando as restrições impostas para a execução das atividades, descritas anteriormente. Estas especificações podem ser representadas pelos autômatos ilustrados na Figura 5.2. A fim de obter um período mais curto de execução do processo descrito, a especificação E_1 estabelece que logo após a finalização da atividade B – *Definição das Ações Corretivas* a atividade C – *Definição das Ações Preventivas* deve ser realizada.

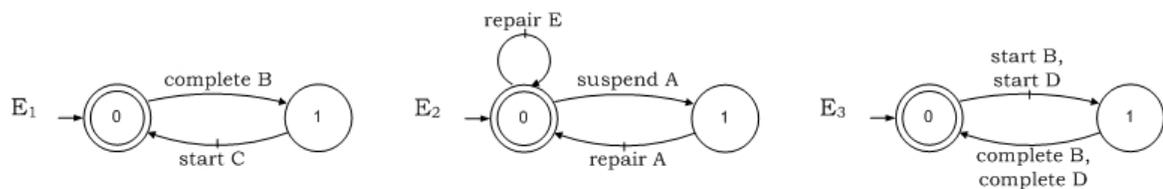


Figura 5.2 - Representação das especificações de controle

A especificação E_1 define que o evento *start C* só pode ocorrer após a ocorrência de um evento *complete B*. A especificação E_2 estabelece que a atividade A – *Identificação das Causas* tem prioridade de reparo, no caso de falhas na execução desta atividade. Isto significa que a reparação da atividade E – *Definição da Proposta de Eficácia*, só pode ser realizada após a reparação da atividade A – *Identificação das Causas*. Por fim, a especificação E_3 estabelece uma exclusão mútua entre as atividades B – *Definição das Ações Corretivas* e D – *Definição dos Por quês*.

De acordo com a abordagem LMC, o primeiro passo para sintetizar supervisores locais é o de obter a planta local para cada especificação. Como o alfabeto contém eventos das atividades e como as atividades não têm eventos comuns, as plantas locais para E_1 , E_2 e E_3 são, respectivamente, dadas por $Gl_1 = G_B | G_C$, $Gl_2 = G_A | G_E$ e $Gl_3 = G_B || G_D$. Estas plantas locais são mostradas na figura 5.3. Os estados das plantas locais estão identificados com os estados correspondentes dos autômatos que lhes deram origem, por meio da operação síncrona. Por exemplo, o estado “00” da planta local Gl_1 corresponde ao estado “0”, tanto em atividades B – *Definição das Ações Corretivas* e C – *Definição das Ações Preventivas*.

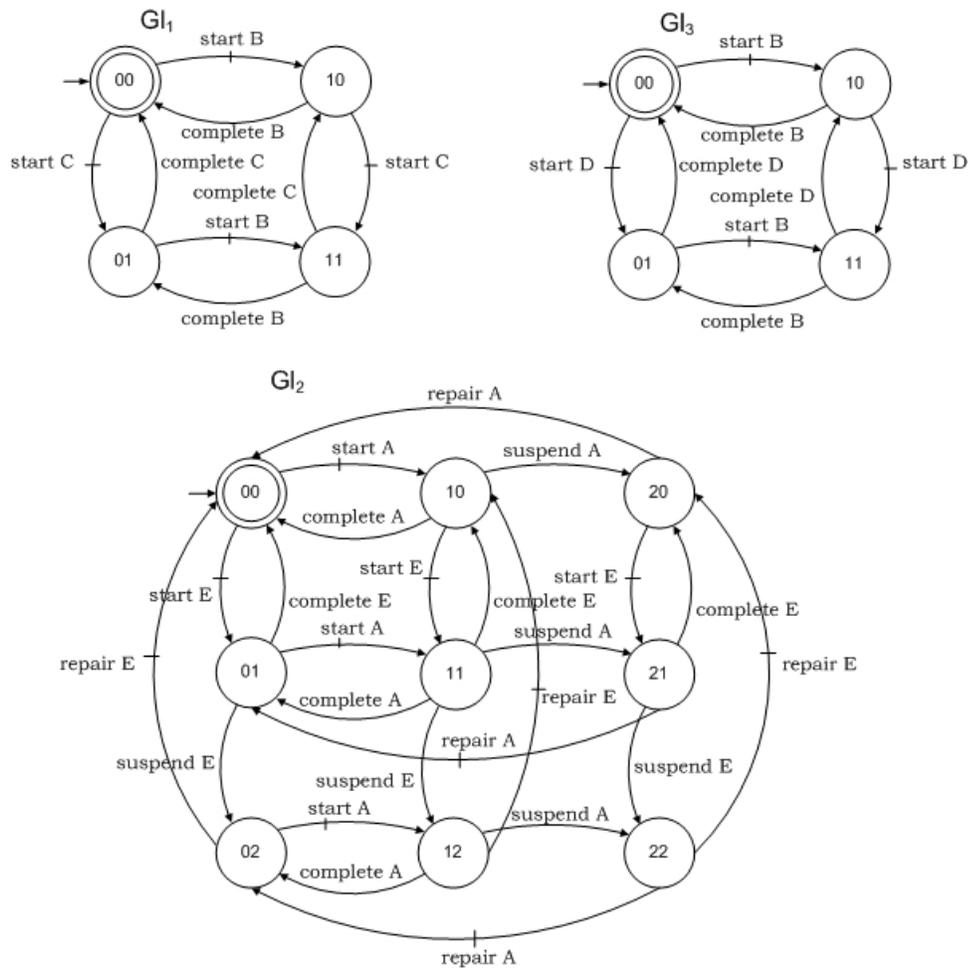


Figura 5.3 - Representação das Plantas Locais

Utilizando os algoritmos propostos por Ramadge e Wonham (1989), é possível obter três supervisores locais, cada um garantindo a satisfação da especificação correspondente. A síntese de um supervisor local S_j é realizada considerando a correspondente especificação E_j e de sua planta local G_j . Ao utilizar este procedimento, é possível sintetizar um supervisor local para cada uma das especificações estabelecidas. O software TCT (FENG e WONHAM 2006) foi utilizado para executar a composição síncrona, a síntese de supervisores e o procedimento de redução de supervisores. Os supervisores obtidos S_1 , S_2 e S_3 correspondem às especificações E_1 , E_2 e E_3 , respectivamente, e são mostrados na figura 5.4.

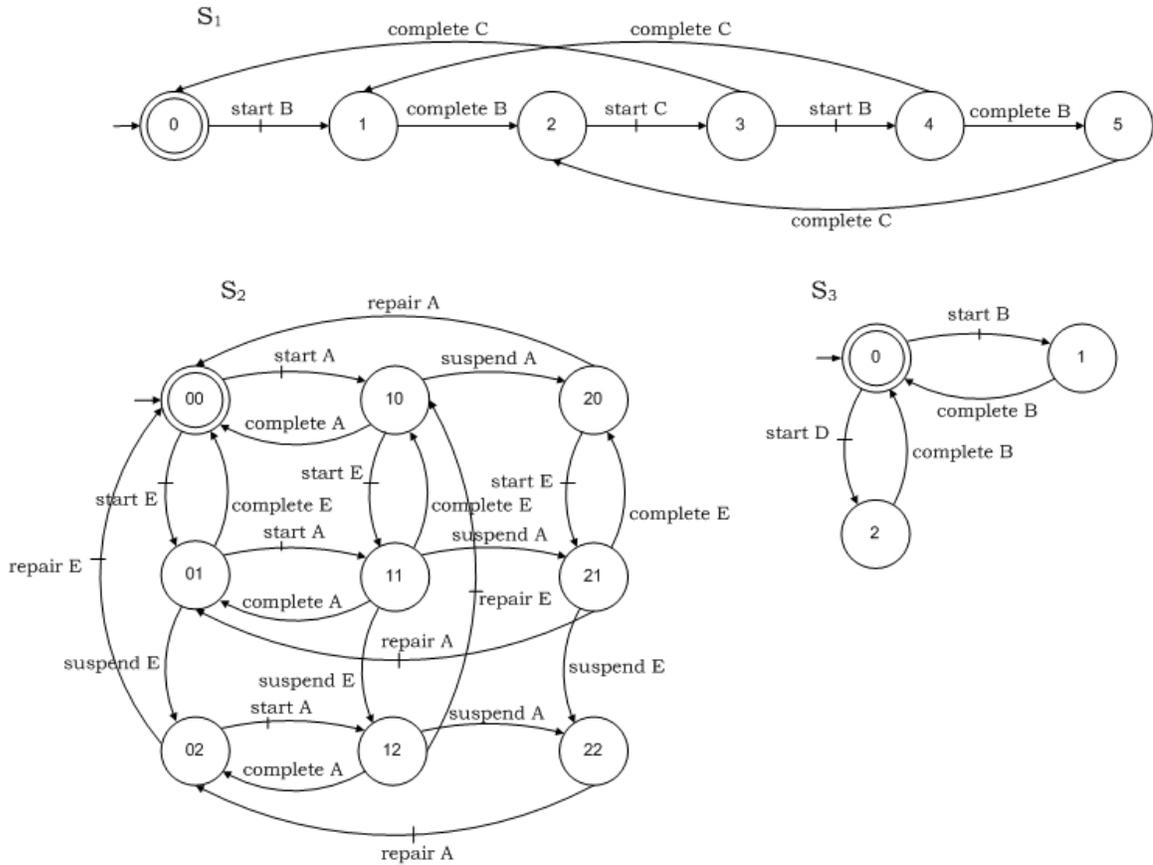


Figura 5.4 - Representação dos Supervisores Locais

Como um passo final antes da implementação física, é levada em consideração a redução dos supervisores, pois esta redução irá possibilitar um número menor de estados de um supervisor, o que pode representar economia de memória, bem como permitir uma lógica de controle mais clara. O supervisor reduzido tem um menor número de estados, porém, possui a mesma ação de controle do supervisor não reduzido correspondente. Por meio do algoritmo de redução proposto por Su e Wonham (2004), para os supervisores S_1 , S_2 e S_3 foram obtidos os supervisores Sr_1 , Sr_2 e Sr_3 . Embora este algoritmo tenha complexidade exponencial, torna-se viável para supervisores locais modulares, que geralmente têm um pequeno número de estados. Cada supervisor poderá ser representado por um par correspondente (S_j, Φ_j) .

Considerando-se os supervisores reduzidos Sr_j ilustrados na figura 5.5, os mapas de saída correspondentes são: para Sr_1 é $\Phi_{r_1}(0) = \text{start C}$, $\Phi_{r_1}(1) = \text{start B}$, para Sr_2 é $\Phi_{r_2}(1) = \text{repair E}$, e para Sr_3 é $\Phi_{r_3}(1) = \text{start B}$, start D . A figura 5.5 ilustra o conjunto de supervisores modulares locais reduzidos, com as ações de

controle, representadas em caixas ligadas ao estado correspondente. Assim, para os três supervisores locais é calculada a ação de controle, atribuindo a cada estado um conjunto de eventos que devem ser desabilitados, ou seja, eventos que, no estado correspondente podem ocorrer na planta local respectiva, porém não são permitidos pelos supervisores.

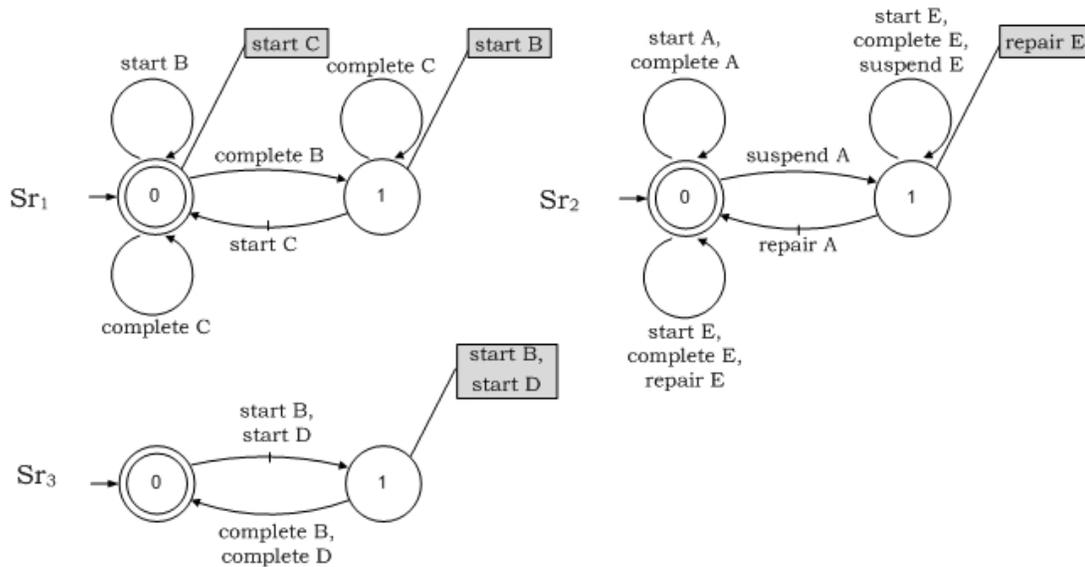


Figura 5.5 - Representação dos Supervisores Locais Reduzidos

Observe que duas atividades estão associadas às atividades *A* e *E*: executar e reparar, o que corresponde aos eventos de início: *start A* e *start E*, e de reparação: *repair A* e *repair E*. Nos estados “0” dos supervisores locais, os usuários podem desempenhar as atividades *A*, *B*, *D* e *E*. Portanto, não é permitida no início a execução da atividade *C*, pois o evento *start C* é desativado pelo supervisor Sr_1 no estado “0”. Após a execução de uma atividade *B* ou *D*, o estado “1” do supervisor Sr_3 é alcançado. Neste momento, os usuários finais estão autorizados a realizar as atividades *B*, *C* e *D*. Sob o ponto de vista do processo e levando em consideração as restrições impostas, é necessário que os usuários realizem todas as atividades de acordo com a seqüência estabelecida pelos supervisores. Porém, dependendo do perfil do usuário no processo, esta seqüência pode ser ignorada, caso seja necessário, flexibilizando desta forma a tomada de decisão pelo usuário, pois os usuários com perfil de Analista de Processo têm conhecimento sobre o cenário do processo de negócio e de quais fatores podem ser relevantes para o melhor resultado na execução do processo em uma situação específica.

No caso das atividades *A* e *E* falharem simultaneamente, o estado “1” do supervisor Sr_2 é atingido e, a atividade de reparação da atividade *E* fica desativada até que a reparação da atividade *A* seja concluída.

A aplicação da abordagem monolítica em um único supervisor global resultou em 90 estados e 401 transições. A representação reduzida correspondente gerou 8 estados e 68 transições. Apesar do fato da aplicação da abordagem monolítica parecer viável neste caso, considerando a redução do supervisor monolítico em um número reduzido de estados, em grandes processos esta aplicação é muito vantajosa. No presente trabalho optou-se em implementar o conjunto de supervisores reduzidos.

5.2 RECURSOS PARA INTEGRAÇÃO DO PAIS COM O SCS

Para possibilitar a comunicação entre o SCS, por meio do OS, e os PAISs utilizados na presente pesquisa, utilizou-se dois recursos tecnológicos: (1) um WebService para permitir a troca dos dados (log de eventos e regras para a execução das atividades) e (2) um Widget desenvolvido para ilustrar as regras retornadas pelo SCS em formato de semáforo, onde as atividades ilustradas com a luz vermelha não devem e/ou não deveriam ser executadas, com base nas regras de negócio e perfil do usuário no processo. A figura D.1 do apêndice D ilustra o WSDL do WebService implementado para a integração entre o PAIS e o SCS.

O WebService implementado possui um método chamado: *readFromClient* que recebe uma seqüência de caracteres do tipo *string*. Esta seqüência de caracteres recebe os dados: das atividades executadas e os respectivos eventos disparados, além do perfil do usuário que executou as atividades no PAIS. Ao receber uma chamada externa, o WebService verifica os dados que foram recebidos, por meio do método *readFromClient* e, se os dados estiverem de acordo com o formato definido, serão encaminhados para o OS no ProM. O OS por sua vez, irá disponibilizar os dados recebidos para os provedores de serviço, e conseqüentemente para o SCS. Optou-se pela implementação de um WebService para a troca de dados entre um PAIS e o SCS, pois o WebService, como já mencionado no capítulo 2, é implementado de forma independente e, portanto, pode

ser utilizado por qualquer sistema de informação. Devido a esta característica do Webservice, foi possível realizar a execução e testes do SCS em dois PAIS diferentes: o YAWL e o Bizagi, conforme demonstração nos tópicos a seguir.

Além do Webservice, também foi implementado um Widget para ilustrar as regras retornadas pelo SCS em formato de semáforo, onde as atividades ilustradas com a luz vermelha não podem ser executadas, seguindo as regras de negócio definidas no SCS. Um Widget é comumente utilizado para apresentar um tipo de serviço por meio da interface gráfica para o usuário. Há algumas implementações de Widget que permitem também a interação do usuário, no entanto, o Widget desenvolvido no presente trabalho tem como objetivo apresentar o guia das atividades que podem ou devem ser executadas no fluxo do processo de negócio flexível, ou seja, não há interação com o usuário, apenas apresentação das informações em forma de guia para que o usuário tenha conhecimento da situação atual do processo. A figura 5.6 ilustra a interface do Widget apresentada para o usuário.

SCS Guia Online



Figura 5.6 – Widget com o guia do SCS

Na figura 5.6 é possível verificar as informações do guia de atividades retornado pelo SCS com base no log de eventos recebido e as regras de negócio do processo previamente definidas no SCS. O log de eventos do PAIS que é enviado para o SCS é representado no Widget pelas informações que estão abaixo da seção: “Dados enviados para o SCS”. O log de eventos é composto dos seguintes dados:

- a) *case*: representa o identificador da instância do processo atual. No caso do processo utilizado, o SGA, o *case* enviado é o código da anomalia gerada na atividade: Identificação da Anomalia;
- b) *evento*: representa o último evento que foi disparado ao executar uma atividade. No exemplo apresentado na figura 5.6, o evento disparado foi o: *start_A*, ou seja, o usuário do processo iniciou a execução da atividade A do processo, a atividade: Identificação das Causas do SGA;
- c) *usuário*: representa os dados do usuário que executou a atividade e originou o evento. Os dados do usuário encaminhados pelo PAIS para o SCS são: Identificador do Usuário, código do usuário registrado no PAIS; Nome do Usuário e Perfil do Usuário configurado no PAIS. No exemplo apresentado na figura 5.6 o perfil do usuário é: *Operator*, ou seja, o usuário tem o perfil de Operador e, portanto, poderá executar somente as atividades habilitadas pelo guia do SCS. As atividades que estão com o farol vermelho, estarão desabilitadas para este tipo de usuário, para que este não viole a regra de negócio previamente definida no SCS. No entanto, se o usuário tivesse o perfil: *Analysis*, Analista do Processo, receberia as informações do guia, porém não teria restrição para a execução das atividades, pois o SCS considera que este perfil tem conhecimento das regras de negócio e, portanto pode utilizar o seu conhecimento para analisar o cenário atual e definir qual a melhor seqüência de execução das atividades do processo, a fim de garantir o melhor resultado para o alcance da meta do processo.

A figura 5.7 ilustra uma visão geral da arquitetura desenvolvida para o SCS de acordo com a abordagem de controle e supervisão abordada no capítulo 3, com o projeto do SCS elaborado e especificado no capítulo 4, além dos recursos implementados para a integração com o PAIS discutidos neste tópico.

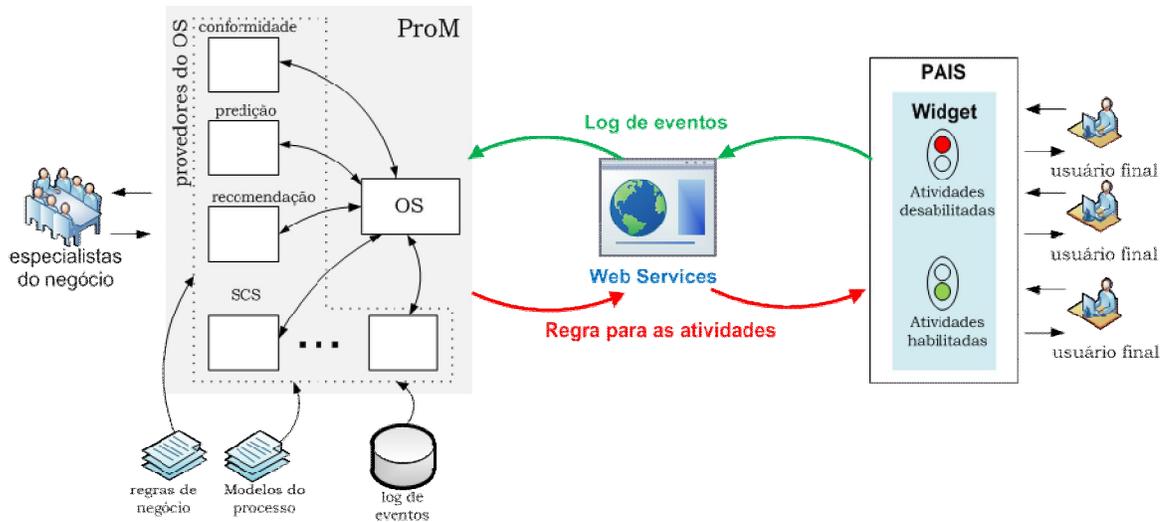


Figura 5.7 – Visão Geral da Arquitetura do SCS

5.3 CONFIGURAÇÃO DO SCS PARA SUPERVISIONAR UM PAIS

Para permitir que o SCS supervisione a execução de um processo de negócio em um PAIS são necessárias as seguintes configurações:

- definir as atividades do processo e os eventos destas atividades que serão supervisionados pelo SCS;
- definir os papéis do processo que possuem acesso de Analista e Operador do Processo;
- especificar as regras de controle dos autômatos nos arquivos XML da arquitetura do SCS.

A definição das atividades e eventos do processo que serão supervisionados pelo SCS deve ser feita no arquivo: *RulesCMLLevel.xml*. A figura 5.8 ilustra o formato do arquivo para a definição das atividades e processos. Todas as atividades do processo de negócio que devem ser monitoradas pelo SCS, bem como os eventos que porventura sejam disparados durante a execução de cada uma das atividades, devem estar configurados neste arquivo, seguindo o formato especificado. A definição das atividades ilustrada na figura 5.8 utiliza o identificador das atividades: A, B, C, D e E, como chave para a identificação da atividade do SGA e dos eventos que esta pode disparar durante a sua execução.

A definição dos papéis do processo e os tipos de permissão que estes possuem, permissão de Analista e/ou Operador, também deve ser configurado no arquivo: *RulesCMLevel.xml*. Neste arquivo, há uma seção nomeada como: *Roles*, onde todos os perfis do processo deverão ser descritos e relacionados a um dos tipos de permissões disponíveis: Analista do Processo ou Operador do Processo. A figura 5.8 ilustra os dois perfis do processo que foram utilizados para o SGA: *Analysis* e *Operator*. Coincidentemente estão sendo utilizados apenas dois perfis de acesso, assim como os tipos de permissão definidos, no entanto, é possível configurar “N” perfis de processo, com base na configuração do processo realizada no PAIS, e relacionar com os tipos de permissão disponíveis no SCS. Quando um perfil de processo estiver definido com o tipo: Analista, os usuários que tiverem este perfil terão a permissão de ignorar a seqüência de atividades definida por meio das restrições impostas pelas regras de negócio do processo. No entanto, quando o perfil de processo estiver definido com o tipo: Operador, os usuários que tiverem este perfil deverão executar as atividades do processo de acordo com o guia retornado pelo SCS, pois o SCS deixará as atividades com o semáforo vermelho e inabilitadas para execução por este usuário.

```

<?xml version="1.0" encoding="UTF-8" ?>
- <supervisory>
- <activities>
- <activity>
- <activity>
  <name>a</name>
  <value>true</value>
- <events>
- <event>
  <name>start_A</name>
  </event>
- <event>
  <name>complete_A</name>
  </event>
- <event>
  <name>fail_A</name>
  </event>
- <event>
  <name>repair_A</name>
  </event>
  </events>
</activity>
- <activity>
  <name>b</name>
  <value>true</value>
+ <events>
</activity>
+ <activity>
+ <activity>
+ <activity>
</activities>
- <roles>
  <role name="Analysis" type="Analysis" />
  <role name="Operator" type="Operator" />
</roles>
</supervisory>

```

Figura 5.8 – Estrutura do arquivo RulesCMLevel.xml

A especificação das regras de controle deve ser feita nos arquivos: *AutomataStructurePSLevel.xml* e *AutomataStructureMSLevel.xml*. Nestes arquivos são descritas as especificações dos autômatos responsáveis pela supervisão local e supervisão centralizada do SCS, com base nas regras de controle definidas no tópico 5.1. A figura 5.9 ilustra a estrutura do arquivo: *AutomataStructureMSLevel.xml* e a especificação para a supervisão centralizada. O formato utilizado para a especificação das regras dos autômatos baseia-se no formato utilizado pela ferramenta Supremica (LJUNGKRANTZ et al. 2007; AKESSON et al. 2006; AKESSON et al. 2003). Esta ferramenta utiliza os conceitos da teoria de controle supervísório (SCT) e permite a construção e especificação de autômatos.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <Automata name="SupervisaoCentralizada">
- <Automaton name="Sr1" type="Specification">
- <Events>
  <Event id="0" label="startB" controllable="false" />
  <Event id="1" label="completeB" controllable="false" />
  <Event id="2" label="completeC" controllable="false" />
  <Event id="3" label="completeC" controllable="true" />
</Events>
- <States>
  <State id="0" name="0" output="startC" initial="true" />
  <State id="1" name="1" output="startB" />
</States>
- <Transitions>
  <Transition source="0" dest="0" event="0" />
  <Transition source="0" dest="1" event="1" />
  <Transition source="1" dest="1" event="2" />
  <Transition source="1" dest="0" event="3" />
</Transitions>
</Automaton>
<Automaton name="Sr2" type="Specification" />
<Automaton name="Sr3" type="Specification" />
</Automata>
```

Figura 5.9 - Especificação das regras no SCS

Após finalizar as configurações do SCS, devem-se iniciar as configurações dos PAISs para que seja possível ao SCS supervisionar o processo durante a sua execução. Estas configurações, no entanto, devem ser feitas somente quando está configurando a supervisão e regras de controle do processo pela primeira vez, ou ainda quando é implantada alguma melhoria no processo.

5.4 CONFIGURAÇÃO DO PAIS

Para que fosse possível realizar o experimento com o SCS, foi necessário realizar a configuração do processo de negócio SGA nos PAISs: Bizagi e YAWL. Em um cenário organizacional, grande parte deste tipo de configuração é dispensável, visto que os processos de negócio que são executados por um PAIS já estão devidamente configurados.

Como o presente trabalho utilizou dois PAISs distintos, a fim de comprovar a aplicabilidade do SCS em PAISs com infra-estrutura e arquitetura diferentes, as configurações são definidas para cada tipo de PAIS. Resumidamente os itens que devem ser observados para a configuração do PAIS para ser supervisionado pelo SCS são:

- a) modelo do processo de negócio flexível devidamente implementado e implantado;
- b) papéis do processo definidos;
- c) usuários do processo relacionados aos papéis do processo;
- d) configuração do Webservice que será utilizado para a troca de informações entre o PAIS e o SCS;
- e) configuração do Widget para ser apresentado nas telas do PAIS;
- f) configuração dos eventos das atividades do processo e habilitação / inabilitação das atividades com base no retorno do SCS.

Estas configurações foram realizadas nos dois PAISs utilizados: o Bizagi e o YAWL, e são descritas no Apêndice do presente trabalho, devido à grande quantidade de informação e passos necessários para a sua realização.

5.5 PLANEJAMENTO E EXECUÇÃO DO SGA

Ao finalizar as configurações do SCS e dos PAISs, elaborou-se um plano de testes com o objetivo de planejar os testes que seriam realizados para a condução do experimento. No plano de testes foram considerados o escopo que seria utilizado para a realização dos testes, além dos tipos de testes que seriam realizados. Também foi considerado o ambiente e os recursos que seriam necessários para a realização dos testes. O plano de testes elaborado é apresentado na tabela E.1 do apêndice E. A elaboração de um plano de testes para a condução do experimento facilita a identificação dos tipos de testes que seriam realizados bem como as abordagens que deveriam ser adotadas para a realização de cada teste. Além disso, este plano de testes poderá ser utilizado futuramente para a evolução deste trabalho.

Com o plano de testes definido, elaboraram-se os casos de testes com os cenários que seriam utilizados para avaliar a aplicação do SCS na supervisão de um processo em execução. Por meio dos casos de testes foi possível detalhar os procedimentos necessários para avaliar o experimento, bem como os resultados esperados após a realização de cada procedimento. Os casos de testes elaborados são apresentados nas tabelas E.2 a E.4 do apêndice E.

Com os casos de testes elaborados, teve início a execução do processo SGA nos PAISs, seguindo os cenários definidos nos casos de testes. Os resultados obtidos durante a execução dos cenários estavam de acordo com os resultados esperados definidos nos casos de testes. A fim de ilustrar os resultados obtidos, nos tópicos a seguir são apresentadas as figuras que demonstram parte da execução dos casos de testes em cada PAIS. No Bizagi, utilizou-se também a regra de restringir a execução das atividades seguindo o guia do SCS, pois o usuário utilizado para a execução das atividades possuía o perfil de Operador do Processo. Já no YAWL, o usuário utilizado possuía o perfil de Analista do Processo e, portanto, não teve as atividades inabilitadas para execução, sendo o guia do SCS apenas um auxílio para apresentar a melhor seqüência de atividades com base nas regras de negócio do processo.

5.5.1 Execução do SCS no Bizagi

Ao iniciar o processo do SGA no Bizagi, as atividades que antecedem as atividades configuradas para a supervisão do SCS são realizadas. Estas atividades estão especificadas no modelo do processo, na seção de Identificação da Anomalia, conforme demonstra a figura A.1 do apêndice A. Após a realização destas atividades, têm início a seção de investigação das causas, seção que possui as cinco atividades supervisionadas pelo SCS. A figura 5.10 ilustra a visão inicial das atividades que estão habilitadas para execução pelo usuário com o perfil de operador.



Figura 5.10–Atividades habilitadas para execução inicialmente no Bizagi

Conforme ilustrado na figura 5.10, a atividade C – Definição das Ações Preventivas não é apresentada para execução, pois esta foi inabilitada pelo SCS, seguindo as regras de controle estabelecidas. A figura 5.11 ilustra o início da execução da atividade B – Definição das Ações Corretivas. A figura apresenta a tela de execução da atividade com o Widget e o guia do SCS.



Figura 5.11–Início da execução da atividade B no Bizagi

O guia do SCS ilustrado por meio da figura 5.11 indica que as atividades: B, C e D estão inabilitadas para execução pelo usuário. As figuras 5.12 e 5.13 apresentam, respectivamente, a execução do SCS no ProM e o resultado obtido por meio do log de execução recebido e a tela com a lista de atividades que podem ser executadas pelo usuário.

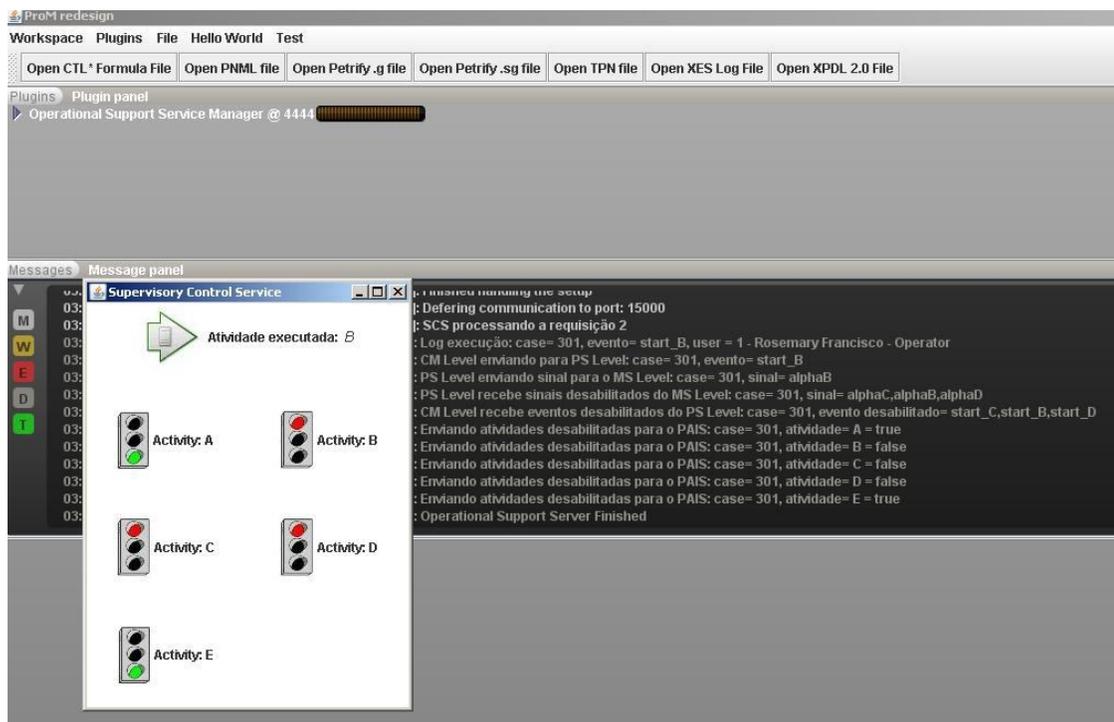


Figura 5.12–Execução do SCS com base na execução da atividade B no Bizagi

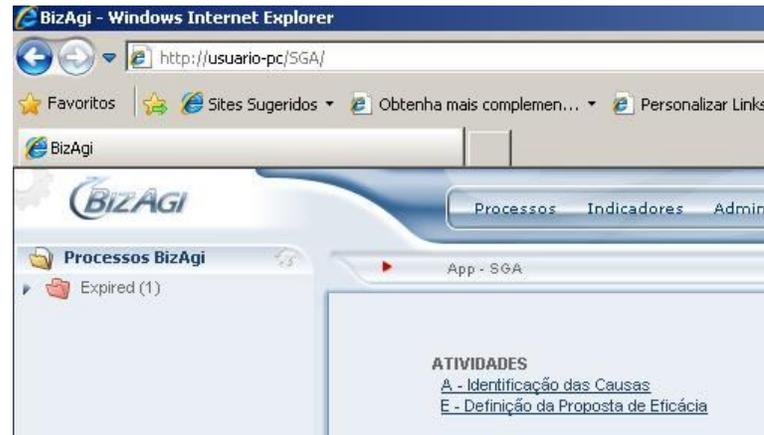


Figura 5.13–Atividades habilitadas para execução durante a execução da atividade B no Bizagi

Ao finalizar a execução da atividade B, o SCS recebe o log de execução indicando a ocorrência do evento: *complete_B*. A figura 5.14 ilustra a execução do SCS após a execução da atividade B. A figura 5.15 ilustra a lista de atividades habilitadas para execução após a finalização da atividade B.

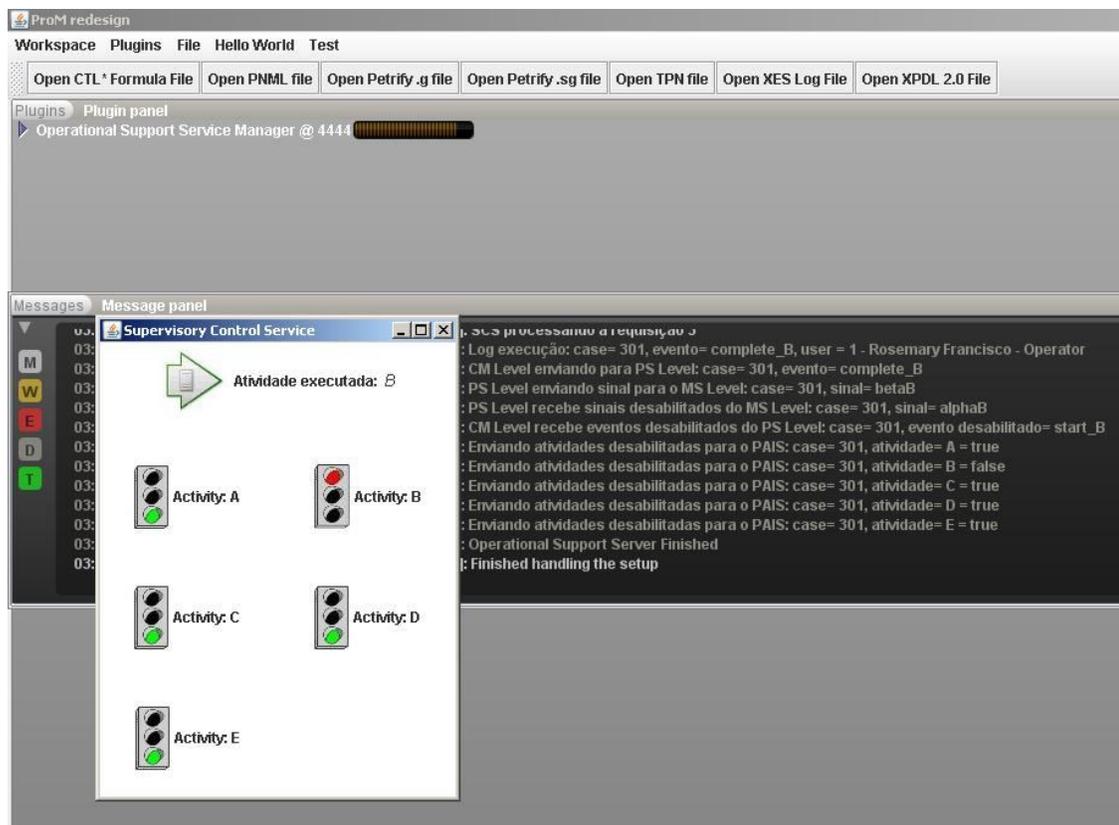


Figura 5.14–Execução do SCS após a finalização da execução da atividade B no Bizagi



Figura 5.15–Atividades habilitadas para execução após a finalização da execução da atividade B no Bizagi

5.5.2 Execução do SCS no YAWL

Assim como no Bizagi, ao iniciar o processo do SGA, as atividades que antecedem as atividades configuradas para a supervisão do SCS são realizadas. Estas atividades estão especificadas no modelo do processo, conforme demonstra a figura B.1 do apêndice B. Após a realização destas atividades, são habilitadas as cinco atividades supervisionadas pelo SCS. A fim de validar a regra de restringir ou não as atividades, com base no perfil do usuário, no YAWL o usuário foi configurado com o perfil de Analista do Processo. Por meio deste perfil, todas as atividades ficam habilitadas para a execução do usuário, sendo o SCS apenas um guia informativo de melhor seqüência de execução das atividades do processo seguindo as regras de negócio estabelecidas. A figura 5.16 ilustra a visão inicial das atividades habilitadas para execução pelo usuário com o perfil de analista.

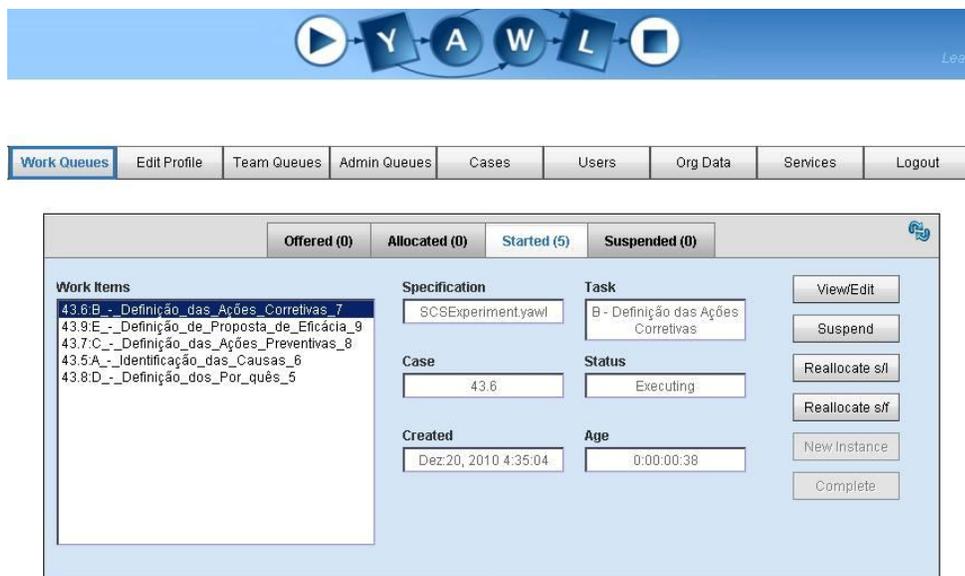


Figura 5.16–Atividades habilitadas para execução inicialmente no YAWL

A figura 5.17 ilustra o início da execução da atividade B – Definição das Ações Corretivas. A figura apresenta a tela de execução da atividade com o Widget e o guia do SCS. Como mencionado na configuração do YAWL no apêndice B, e pode ser observado na figura 5.17, no YAWL é possível implementar um layout personalizado para a execução das atividades do processo. Este recurso foi utilizado para a apresentação das informações da atividade e do Widget do SCS.



Figura 5.17–Início da execução da atividade B no YAWL

O guia do SCS ilustrado por meio da figura 5.17 se comporta da mesma forma que no Bizagi. O guia indica que as atividades: B, C e D estão inabilitadas para execução pelo usuário segundo as regras de negócio estabelecidas. No entanto, como o usuário possui perfil de analista do processo, as atividades continuam habilitadas para o usuário para execução, caso este usuário identifique que para esta instância do processo específica, uma alteração na seqüência das atividades teria melhor resultado para o processo. A figura 5.18 apresenta a execução do SCS no ProM e o resultado obtido por meio do log de execução recebido.

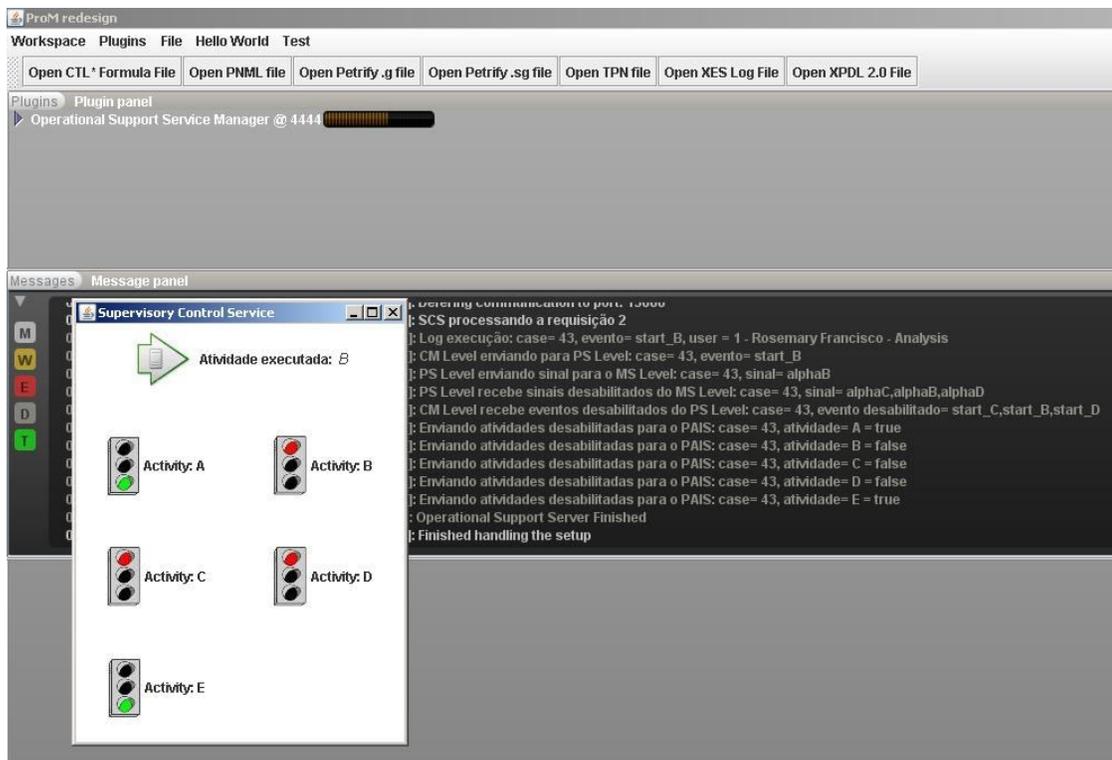


Figura 5.18—Execução do SCS com base na execução da atividade B no YAWL

Ao finalizar a execução da atividade B, o SCS recebe o log de execução indicando a ocorrência do evento: *complete_B*. A figura 5.19 ilustra a tela do YAWL com a indicação de finalização da atividade B. A figura 5.20, por sua vez, ilustra a execução do SCS após a execução da atividade B.



Figura 5.19—Finalização da execução da atividade B no YAWL

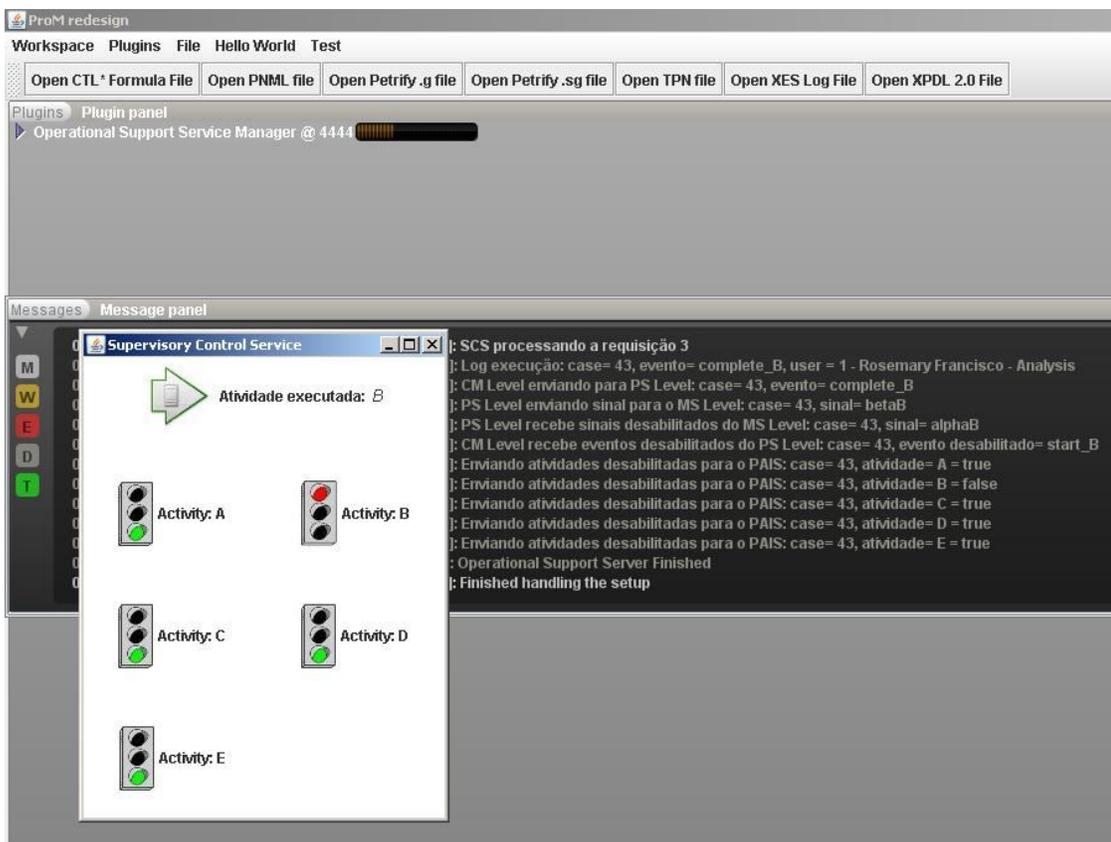


Figura 5.20—Execução do SCS após a finalização da execução da atividade B no YAWL

Por fim, a figura 5.21 ilustra a lista de atividades habilitadas para execução pelo usuário no YAWL, após a execução da atividade B.

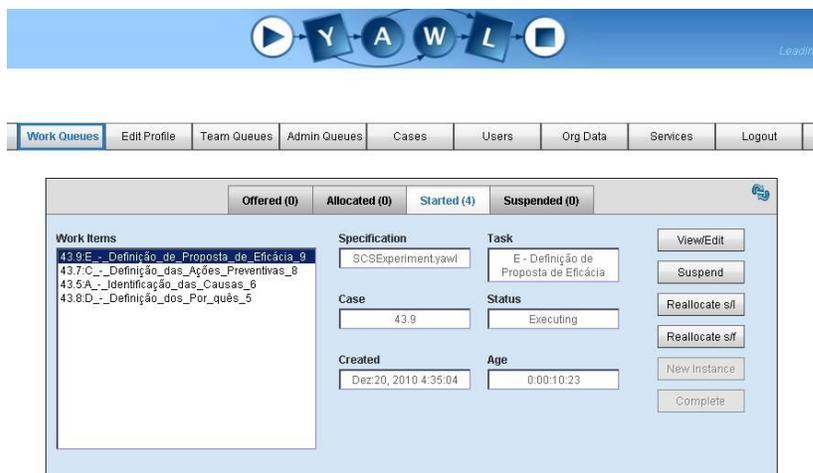


Figura 5.21–Atividades habilitadas para execução após a finalização da execução da atividade B

5.6 CONCLUSÃO DO CAPÍTULO

Neste capítulo foram apresentadas as definições para as regras de controle do SCS no exemplo de modelo de processo adotado para conduzir o experimento, além das configurações e procedimentos necessários para a realização do experimento para validação da abordagem proposta pelo presente trabalho. Devido a grande quantidade de informação e configurações, parte do detalhamento das configurações, referente a configuração dos PAISs, foi disponibilizado nos apêndices. Tanto as configurações do SCS quanto as configurações dos PAISs foram imprescindíveis para a condução do experimento e para o alcance dos resultados esperados. A fim de auxiliar na condução do experimento, elaborou-se também um plano de testes que permitiu especificar o ambiente necessário para os testes bem como os tipos de testes e abordagem utilizada para a observação dos resultados. Também foram elaborados três casos de testes que possibilitaram o detalhamento dos cenários de testes e os procedimentos que deveriam ser utilizados para validar a funcionalidade do SCS ao supervisionar os PAISs. Os testes foram realizados com base nos cenários definidos e os resultados esperados foram alcançados. Algumas imagens que demonstram o sucesso dos testes e do experimento nos dois PAISs, também foram apresentadas neste capítulo.

6 CONCLUSÕES

A Gestão por processos possibilita às organizações um enfoque administrativo para otimizar e melhorar a sua cadeia de processos, além de atender as necessidades e expectativas das partes interessadas. Utilizando as técnicas de BPM é possível unir as melhores práticas e ferramentas da Gestão por Processos e da Tecnologia da Informação com o intuito de identificar, mapear e implementar melhorias nos processos de negócio de maneira mais ágil e continuada. Além disso, ao utilizar as ferramentas BPM como suporte aos processos, será possível identificar as mudanças que ocorrem no ambiente dos negócios, de maneira mais rápida e implementar as modificações necessárias nestes processos a fim de produzir os meios para as organizações tornarem-se ou manterem-se competitivas.

Ao utilizar um PAIS flexível, as organizações podem usufruir ainda mais da característica de capacidade de mudança com base na demanda. Porém, com base nos estudos realizados, identificou-se que PAIS flexíveis exigem usuários mais experientes para a condução dos seus processos de negócio. Portanto, um mínimo de controle deve ser considerado, visto que restrições críticas podem impactar em prejuízos e distanciar as organizações da almejada posição de competitividade. Sabe-se que, quanto maior a complexidade envolvida em um processo de negócio, maior deve ser a necessidade de se desenvolver a capacidade de gestão destes processos. Uma solução que se mostrou viável por meio desta pesquisa é a viabilização de um PAIS flexível com suporte integrado para apoio aos usuários. Neste tipo de aplicação, a flexibilidade para a tomada de decisão é mantida, porém o suporte baseado nas regras de negócio do processo é disponibilizado.

O SCS foi então implementado no ProM e as aplicações nos softwares YAWL e Bizagi mostrou a factibilidade da proposta. A infra-estrutura computacional anteriormente implementada no ProM permitiu a comunicação com aplicações externas, e o SCS pode ser integrado ao YAWL e ao Bizagi, com o apoio de um Webservice utilizando os principais conceitos de SOA. Além disso, foi implementada uma interface de comunicação com os usuários do processo, um Widget, que permitiu a apresentação visual das atividades habilitadas / desabilitadas com base nos eventos ocorridos e nas regras de negócio preestabelecidas para a

execução do processo. A estrutura do SCS e os recursos implementados viabilizaram a realização dos testes nos dois PAISs, independente da infra-estrutura e arquitetura utilizada por estes e comprovaram a validade da abordagem proposta.

6.1 AVALIAÇÃO DE RESULTADOS

Seguindo os objetivos específicos definidos, a primeira etapa da pesquisa teve como parâmetro revisar e fundamentar os principais conceitos envolvidos na pesquisa. Esta fundamentação estabeleceu a base para a definição do tema e problema de pesquisa, bem como a idéia principal da pesquisa. O segundo objetivo específico estabeleceu como meta a elaboração e definição da estrutura do serviço de controle supervisorio (SCS) para possibilitar o monitoramento e controle do processo e suas regras de negócio. Este objetivo também foi alcançado com base na fundamentação do conceito de teoria de controle supervisorio e permitiu a elaboração da estrutura principal do projeto de pesquisa. Com a estrutura do projeto definida, iniciou-se a análise dos métodos de pesquisa a fim de selecionar um método mais adequado às características do projeto de pesquisa. Definiu-se neste momento que a pesquisa experimental estaria adequada aos critérios da pesquisa possibilitando a aplicação e análise dos resultados na forma de um experimento de laboratório.

O terceiro objetivo estabeleceu a implementação do serviço de controle supervisorio projetado de acordo com a estrutura definida. O SCS foi desenvolvido como um provedor da plataforma do ProM, portanto foi necessário estudar a plataforma, identificar os padrões e adequar os itens necessários a fim de garantir o perfeito funcionamento do SCS como um provedor de serviço no ProM. Existem duas formas de implementação de recursos no ProM, o desenvolvimento de plug-ins e o desenvolvimento de provedores de serviço. Este último caracteriza-se por seu controle estar atrelado a um plug-in. No caso específico do SCS, seu funcionamento ficou atrelado ao plug-in Operational Support (OS). Por este motivo, também foi necessário o estudo da estrutura e arquitetura do OS, para que o controle no envio e recebimento das informações fosse permitido.

O quarto objetivo definiu a implementação dos recursos que permitiriam a integração do serviço de controle supervisorio com as ferramentas PAISs, possibilitando desta forma que o serviço pudesse monitorar os eventos e retornar o suporte necessário para os usuários durante a execução do processo. O resultado deste objetivo foi expressivo para a realização dos testes a fim de validar a aplicabilidade da abordagem proposta. Tanto o YAWL quanto o Bizagi se mostraram ferramentas adequadas para os testes, principalmente por serem ferramentas BPMS e por permitirem integração com outros processos e ou aplicativos externos por meio das ferramentas de SOA. A vantagem da ferramenta BPMS em relação a outras ferramentas é a disponibilização de recursos para todo o ciclo de vida dos processos. Para o alcance deste objetivo foi então implementado um Webservice para conexão com o SCS por qualquer ferramenta externa. A utilização do Webservice viabiliza a integração com ferramentas heterogêneas e permite a troca das informações de maneira transparente. Além disso, para que fosse possível apresentar aos usuários a situação atual do processo e a aplicação das regras de negócio durante a execução do processo, foi implementado um Widget para a apresentação gráfica do guia de execução do processo retornado pelo SCS com a indicação das atividades habilitadas / inabilitadas para a execução pelos usuários. Este guia apresentado de forma gráfica permite aos usuários conhecerem as regras de negócio aplicadas ao processo e, com o tempo, possibilitar a formação de uma opinião crítica acerca das regras aplicadas, para que seja possível a identificação de possíveis pontos de melhorias.

Por fim, o quinto e último objetivo teve como premissa a elaboração do plano e casos de testes que permitiram planejar uma seqüência lógica para os testes e a validação dos resultados. Por meio do plano de testes elaborado foi possível planejar o escopo dos testes que seriam realizados, além dos tipos e abordagem de testes que seriam aplicadas com base no escopo definido. O ambiente do teste também foi definido no plano de testes, sendo este tipo de informação imprescindível para a reprodução dos testes realizados em outros contextos e/ou cenários. Os cenários utilizados para a condução dos testes foram especificados em três casos de testes. Nestes casos de testes foram detalhados os procedimentos que deveriam ser realizados, além dos dados de entrada e resultados esperados com base nas regras de negócio do processo em observação.

O objetivo geral da pesquisa: “propor um serviço de controle e supervisão que possibilite manter a flexibilidade do PAIS e também permitir o suporte aos usuários, no formato de um guia em tempo real, durante a execução dos processos de negócio suportados por um sistema de informação (PAIS)” foi então atendido por meio da concretização de todos os objetivos específicos.

6.2 SUGESTÃO PARA TRABALHOS FUTUROS

Trabalhos futuros irão investigar a aplicação e implementação do SCS em processos de negócio flexíveis de grande porte. Objetiva-se fazer do SCS uma importante ferramenta para o suporte a decisão na execução de sistemas de informação que suportam os processos. Para que esta aplicação seja possível é importante o desenvolvimento de uma interface mais amigável e intuitiva que permita aos analistas dos processos configurar e/ou reconfigurar as regras de negócio no SCS de maneira rápida. Atualmente esta configuração é realizada em arquivos XML e exige um usuário com um nível de conhecimento mais avançado para a configuração das regras neste formato de arquivo.

Outro item interessante que pode ser implementado no SCS refere-se ao versionamento das regras de negócio do processo. Este tipo de recurso permitiria a manutenção de um histórico das regras de negócio aplicadas aos processos, e o controle dos analistas de processos que fizeram as alterações. A manutenção deste histórico permitiria análises posteriores sobre o comportamento do processo e as regras utilizadas, bem como a recuperação de uma regra que foi alterada indevidamente ou mesmo que poderia ser reaplicada devido ao resultado proporcionado ao processo durante a sua utilização.

O desenvolvimento de um repositório para o armazenamento dos logs de eventos monitorados pelo SCS também pode ser analisado como uma futura implementação. Este recurso permitiria salvar os logs de eventos no formato MXML, formato utilizado pelo ProM Framework, e permitiria registrar as atividades executadas para análises posteriores, como por exemplo: quais restrições foram violadas pelos analistas de processos, qual foi o impacto destas violações, etc.

Ainda como sugestão pode-se considerar a utilização de outras ferramentas PAIS para a integração com o SCS, considerando os diversos tipos de ferramentas PAIS existentes. As ferramentas utilizadas na presente pesquisa são ferramentas classificadas como P2A e são do tipo BPMS, sendo estas já preparadas para a forma de integração desenvolvida neste trabalho. Desta forma, sugere-se a utilização do SCS com outros tipos de PAIS, como os PAIS classificados como P2P e A2A para a evolução do serviço de controle e supervisão para uma abrangência maior na integração com outros tipos de ferramentas PAIS.

REFERÊNCIAS

AALST, W.M.P. **Process-Aware Information Systems: Lessons to Be Learned from Process Mining**. In Transactions on Petri Nets and Other Models of Concurrency II, volume 5460 of Lecture Notes in Computer Science, pages 1-26. Springer-Verlag, Berlin, 2009a.

AALST, W.M.P. **Process-Aware Information Systems: Design, Enactment and Analysis**. In B.W. Wah, editor, Wiley Encyclopedia of Computer Science and Engineering, pages 2221-2233. Wiley & Sons, 2009b.

AALST, W.M.P.; PESIC, M.; SCHONENBERG, H. **Declarative Workflows: Balancing Between Flexibility and Support**. Computer Science - Research and Development, 23(2):99-113, 2009.

AALST, W.M.P.; ROSEMAN, M.; DUMAS, M. **Deadline-based escalation in process-aware information systems**. Decision Support Systems 43 (2), 492–511, 2007.

AALST, W.M.P.; DONGEN, B.F.; GÜNTHER, C.W.; MANS, R.S.; DE MEDEIROS, A.K.A.; ROZINAT, A.; RUBIN, V.; SONG, M.; VERBEEK, H.M.W.; WEIJTERS, A.J.M.M. **ProM 4.0: Comprehensive Support for Real Process Analysis**. In: Kleijn, J., Yakovlev, A. (eds.) ICATPN 2007. LNCS, vol. 4546, pp. 484–494. Springer, Heidelberg (2007).

AALST, W.M.P.; WESKE, M.; GRÜNBAUER, D. **Case handling: a new paradigm for business process support**. Data Know Eng 53(2):129–162, 2005.

AALST, W.M.P.; WEIJTERS, A.J.M.M.; MARUSTER, L. **Workflow Mining: Discovering Process Models from Event Logs**. IEEE Transactions on Knowledge and Data Engineering, 16(9):1128–1142, 2004.

AALST, W.M.P.; MEDEIROS, A.K.A. **Process Mining and Security: Detecting Anomalous Process Executions and Checking Process Conformance**. In N. Busi, R. Gorrieri, and F. Martinelli, editors, Second International Workshop on Security Issues with Petri Nets and other Computational Models (WISP 2004), pages 69-84. STAR, Servizio Tipografico Area della Ricerca, CNR Pisa, Italy, 2004.

AALST, W.M.P. **Challenges in Business Process Management: Verification of business processes using Petri nets**. Bulletin of the EATCS, 80:174-198, 2003.

AALST, W.M.P.; HOFSTEDE, A.H.M.; WESKE, M. **Business Process Management: A Survey**. In W.M.P. van der Aalst, A.H.M. terHofstede, and M. Weske, editors, International Conference on Business Process Management (BPM 2003), volume 2678 of LectureNotes in Computer Science, pages 1-12. Springer-Verlag, Berlin, 2003.

AALST, W.M. P.;HEE, K. M. V.;**Workflow Management: Models, Methods and Systems**.MIT Press. Cambridge, Massachusetts, London, England, 2002.

AALST, W.M. P.**Making Work Flow: On the Application of Petri nets to Business Process Management**. In J. Esparza and C. Lakos, editors, Application and Theory of Petri Nets 2002, volume 2360 of Lecture Notes in Computer Science, pages 1-22. Springer-Verlag, Berlin, 2002.

ADAMS, M.; HOFSTEDE, A.H.M.; EDMOND, D.; AALST, W.M.P. **Worklets: A service-oriented implementation of dynamic flexibility in workflows**. In: Meersman R, Tari Z et al (eds) On the Move to Meaningful Internet Systems 2006, OTM Confederated International Conferences, 14th International Conference on Cooperative Information Systems (CoopIS 2006), vol 4275 of Lecture Notes in Computer Science, pp 291–308. Springer-Verlag, Berlin, 2006.

AGOSTINI, A.; DE MICHELIS, G. **Improving flexibility of workflow management systems**. In: van der AalstWMP, Desel J, Oberweis A (eds) Business process management: Models, techniques,and empirical studies, vol 1806 of Lecture Notes in ComputerScience, pp 218–234. Springer-Verlag, Berlin, 2000.

AKESSON, K.; FABIAN, M.; FLORDAL, H.; MALIK, R. **Supremica - an integrated environment for verification, synthesis and simulation of discrete event systems**. In Proc. IEEE 8th International Workshop on Discrete Event Systems (WODES'06), Ann Arbor, MI, USA, July 2006, pp. 384-385.

AKESON, K.; FABIAN, M.; FLORDAHL, H.; VAHIDI, A. **Supremica - A Tool for Verification and Synthesis of Discrete Event Supervisors**. In Proc. of the 11th Mediterranean Conference on Control and Automation, Rhodes, Greece, 2003.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS (ABNT). **NBR ISO 9000:2005: vocabulário e fundamentos**. Rio de Janeiro: ABNT, 2005.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS (ABNT). **NBR ISO 9001: Sistema de Gestão da Qualidade - Requisitos - Apresentação**. Rio de Janeiro: 2000.

BERTO, R.; NAKANO, D. **Metodologia da pesquisa e a engenharia de produção**. In: ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, 18, Niterói, 1998. Anais. Niterói: UFF/ABEPRO, 1998.

BPMI – Business Process Management Initiative. **Business Process Modeling Notation – BPMN**. Disponível em: <<http://www.bpmi.org>>. Acesso em: 20 jul. 2009.

BOOTH, D. et. al. **W3C Working Group Note 11: Web Services Architecture**. World Wide Web Consortium (W3C), February 2004. Disponível em: <http://www.w3.org/TR/ws-arch/#stakeholder>. Acesso em: 20/07/2009.

BRANDIN, B. A.; MALIK, R.; MALIK, P. **Incremental Verification and Synthesis of Discrete-Event Systems Guided by Counter Examples Transactions** on Control Systems Technology IEEE Vol. 12, Nº 3, May, 2004.

CALEGARE, Álvaro José de Almeida. **Introdução ao delineamento de experimentos**. São Paulo: E. Blücher, 2001. 130 p. ISBN 85-212-0287-3.

CARLSEN, S.; KROGSTIE, J.; SOLVBERG, A.; LINDLAND, O. I. **Evaluating Flexible Workflow Systems**, Hawaii International Conference on System Sciences (HICSS-30), Maui, Hawaii, (1997).

CARROL, J.; LONG, D. **Theory of finite automata**. Prentice-Hall International Editions, 1989.

CASSANDRAS, C.G.; LAFORTUNE, S. **Introduction to discrete event systems**. Kluwer Academic Publishers, Second Edition, 2008

CASSANDRAS, C G.; LAFORTUNE, S. **Introduction to discrete Event Systems**, Kluwer Academic Publishers, 1999.

CHEN, Y.L.; LAFORTUNE, S.; e LIN, F. **Modular supervisory control with priorities for discrete event systems**. In Proceedings of the 34th IEEE Conference On Decision and Control, pp 409-415. 1995.

CLETO, Marcelo G. **A Gestão da Produção nos últimos 45 anos**. Revista FAE Business, n.4. dez., 2002.

CRUZ, Tadeu. **O teatro Organizacional - Construindo e implantando Processos de Negócio**. Rio de Janeiro : E-papers, 2006.

CURY, J.E.R. **Teoria de Controle Supervisório de Sistemas a Eventos Discretos**. V Simpósio Brasileiro de Automação Inteligente, Canela – RS, Brasil, 2001.

DE SORDI, José Osvaldo. **Gestão de processos: uma abordagem da moderna administração**. 2.ed. São Paulo: Saraiva, 2008.

DONGEN, B.; MEDEIROS, A.; VERBEEK, H.; WEIJTERS, A.J.M.M.; AALST, W.M.P. **The ProM framework: A new era in process mining tool support**. Application and Theory of PetriNets 3536, 444–454, 2005.

DUMAS, M.; AALST, W.M.P.; HOFSTEDE, A. **Process-aware information systems: bridging people and software through process technology**. Wiley-Blackwell, 2005.

ELLIS, C.A.; KEDDARA, K.; ROZENBERG, G. **Dynamic change within workflow systems**. In: Comstock N, Ellis C, Kling R, Mylopoulos J, Kaplan S (eds) Proceedings of the Conference on Organizational Computing Systems, pp 10–21, Milpitas, California, ACM SIGOIS, ACM Press, New York, 1995.

FENG, L.; WONHAM, W. **TCT: A computation tool for supervisory control synthesis**. In International Workshop on Discrete Event Systems, pp. 388–389, 2006.

GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo: Atlas, 2002. 175 p. ISBN 978-85-224-3169-4 (broch.).

GONÇALVES, José Ernesto Lima. **As empresas são grandes coleções de processo**. RAE Revista de Administração de Empresas, v. 40, n. 1, p. 6-19, jan./mar. 2000.

HARRINGTON, H. James; ESSELING, Erik K.; NIMWEGEN, Harm van. **Business Process Improvement Workbook: Documentation, Analysis, Design, and**

Management of Business Process Involvement. New York, NY: McGraw-Hill, 1997.

HO, Y. **Scanning the Issue: Dynamics of Discrete Event Systems.** Proceedings of the IEEE, vol. 77, n. 1, pp. 3-6, jan., 1989.

HUMPHREY, J. **Opportunities for SME's in developing countries to upgrade in a global economy.** Working Paper No. 43, International Labour Office, Geneva. 2003.

JOST, W.; SCHEER, A.W. **Business Process Management: A Core Task for any Company Organization.** In: SCHEER, August - Wilhelm et al. Business Process Excellence. New York: Springer, 2002.

KERLINGER, F.N. **Metodologia da pesquisa em Ciências Sociais.** São Paulo: Editora Pedagógica e Universitária, 1979.

KIRCHMER, M. **Business Process Excellence: Enable Through SOA.** In: Business Process Excellence, Rio de Janeiro, Anais. Rio de Janeiro: IDS-Scheer. Volume Único. P.1-42.14 jul. 2006. CD-ROM.

LAURINDO, F.J.B.; CARVALHO, M.M.; PESSÔA, M.S.P.; SHIMIZU, T. **Selecionando uma Aplicação de Tecnologia da Informação com Enfoque na Eficácia: um Estudo de Caso de um Sistema para PCP.** Revista G&P: Gestão e Produção, Vol.9, n.3, p.377-396, São Carlos, edição de dezembro 2002.

LJUNGKRANTZ, Oscar; AKESSON, Knut; RICHARDSSON, Johan; ANDERSSON, Kristin. **Implementing a Control System Framework for Automatic Generation of Manufacturing Cell Controllers.** ICRA 2007: 674-679. 2007.

LIN, R.F.; YANG, M.C.; PAI, Y.H. **A generic structure for business process modeling.** Business Process Management Journal, vol. 8, No 1, pp 19-41, 2002.

MAGALHÃES, A. et al. **Uma estratégia para a gestão integrada de processos e tecnologia da informação através da modelagem de processos de negócio em organizações.** MP2Tec – Núcleo de Pesquisa e Prática em Tecnologia da Universidade do Rio de Janeiro: UNIRIO, 2004.

MEALY, G. H. **A Method to Synthesizing Sequential Circuits.** In: Bell Systems Technical Journal. pp. 1045–1079, 1955.

MOREIRA, Daniel Augusto. **O método fenomenológico na pesquisa**. São Paulo: Pioneira Thomson, 2002.

MUEHLEN, M.Z.; HO, D.T. **Risk Management in the BPM Lifecycle**. In: Third International Conference of Business Process Management. Nancy. BPM. Volume único, p. 77 a 86. 5 a 7 set. 2005

OLIVEIRA, Saulo Barbará de. **Gestão por processos: fundamentos, técnicas e modelos de implementação: foco no sistema de gestão de qualidade com base na ISO 9000:2005 e 9001:2008**. Rio de Janeiro: Qualitymark, 2008.

PAIM, Rafael; CARDOSO, Vinícius; CAULLIRAUX, Heitor; CLEMENTE, Rafael. **Gestão de processos: pensar, agir e aprender**. Porto Alegre: Bookman, 2009.

PESIC, M. **Constraint-based workflow management systems: Shifting control to users**. Phd thesis, Eindhoven University of Technology, Eindhoven, 2008.

QUEIROZ, M.H., CURY, J.E.R. **Synthesis and implementation of local modular supervisory control for a manufacturing cell**. In: Proc. IFAC 6th International Workshop in Discrete Event Systems, Kluwer Academic Publishers, Zaragoza, Spain, 2002.

QUEIROZ, M.H.; CURY, J.E.R. **Modular supervisory control of large scale discrete-event systems**. In: Proc. IFAC 5th International Workshop on Discrete Event Systems: Analysis and Control, Kluwer Academic Publishers, pp. 103-110, 2000.

RAMADGE, P.J.; WONHAM, W.M. **The control of discrete event systems**. In: Proceedings of IEEE, Special Issue on Discrete Event Dynamic Systems, 77:81-98, 1989.

RAMADGE, P.; WONHAM, W. **Supervisory Control of a Class of Discrete Event Processes**. SIAM Journal Control and Optimization pp. 206-230, 1987.

RINDERLE, S.; REICHERT, M.; DADAM, P. **Correctness criteria for dynamic changes in workflow systems: a survey**. Data Knowl Eng 50(1):9-34, 2004.

ROZINAT, A.; AALST, W. M. P. **Conformance checking of processes based on monitoring real behavior**. Inf. Syst., 33(1):64-95, 2008.

SCHONENBERG, H.; MANS, R.; RUSSELL, N.; MULYAR, N.; AALST, W.M.P. **Process flexibility: A survey of contemporary approaches.** In Advances in Enterprise Engineering I: 1st International Workshop Ciao! and 4th International Workshop Eomas at Caise 2008, Montpellier, France, June 16-17, 2008, Proceedings, pp. 16. Springer, 2008a.

SCHONENBERG, H.; MANS, R.; RUSSELL, N.; MULYAR, N.; AALST, W.M.P. **Towards a taxonomy of process flexibility.** In CAiSE Forum, pp. 81–84, 2008b.

SCHONENBERG, H.; Weber, B.; DONGEN, B.; AALST, W.M.P. **Supporting Flexible Processes through Recommendations Based on History.** In Proceedings of the 6th International Conference on Business Process Management, pp. 51–66. Springer, 2008c.

SMITH, H.; FINGAR, P. **Business Process Management: the third wave.** 1. ed. Tampa: Meghan Kiffer Press, 2003.

SRINIVASAN, L. and TREADWELL, J. **An Overview of Service-oriented Architecture, Web Services and Grid Computing.** HP Software Global Business Unit, 2005.

SU, R.; WONHAM, W.M. **Supervisor Reduction for Discrete-Event Systems.** Discrete Event Dynamic Systems, 14(1), pp. 31-53, 2004.

THIOLLENT, M. **Metodologia da Pesquisa-ação.** 8a ed. São Paulo: Cortez, 1998.

TSITSIKLIS, J.N. **On The Control of Discrete-Event Dynamical Systems.** MIT, Cambridge, Technical Report # LIDS-P-1661. 1987.

VERNE, L. **BPM: The Promise and Challenge.** v2., n.1. USA: DSP, 2004.

WAINER, J.; DE LIMA BEZERRA, F. **Constraint-based flexible workflows.** In: Proceedings of the 9th International Workshop on Groupware: Design, Implementation, and Use (CRIWG 2003), vol 2806, pp 151–158 Springer-Verlag, Berlin, 2003.

WEBER, B.; REICHERT, M.; RINDERLE-MA, S. **Change patterns and change support features—enhancing flexibility in process-aware information systems.** Data & knowledge engineering 66 (3), 438–466, 2008.

WESKE, M. **Business Process Management: Concepts, Languages, Architectures.** 368 p. 265 illus., Hardcover; ISBN: 978-3-540-73521-2 Springer-Verlag Berlin Heidelberg 2007.

WESKE, M. **Formal foundation and conceptual design of dynamic adaptations in a workflow management system.** In: Sprague R (ed) Proceedings of the Thirty-Fourth Annual Hawaii International Conference on System Science (HICSS-34). IEEE Computer Society Press, Los Alamitos, 2001.

WONG, K. C.; THISTLE, J. G.; HOANG, H.H.; MALHAMÉ, R. P. **Conflict resolution in modular control with application to feature interaction.** In Proceedings of the 34th IEEE Conference On Decision and Control, pp 416-421. 1995.

WONG, K.; WONHAM, W. **Hierarchical control of discrete-event systems.** Discrete Event Dynamical Systems 6: 241-273. 1996.

WONG, K. C.; WONHAM, W. M. **Modular Control and Coordination of Discrete-Event Systems.** Discrete Event Dynamic Systems, 8(3):241-273. 1998.

ZHONG, H.; WONHAM, W. **On the consistency of hierarchical supervision.** Discrete-event systems, IEEE Transactions on Automatic Control 35(10): 1125-1134. 1990.

APÊNDICES

APÊNDICE A – CONFIGURAÇÃO DO PAIS BIZAGI

Conforme ilustra a figura A.1, o modelo do processo de negócio SGA foi implementado no Bizagi utilizando a notação BPMN. O Bizagi possui uma interface de configuração bem intuitiva e amigável e permite que todos os passos para a implementação de um processo de negócio sejam realizados de forma rápida.

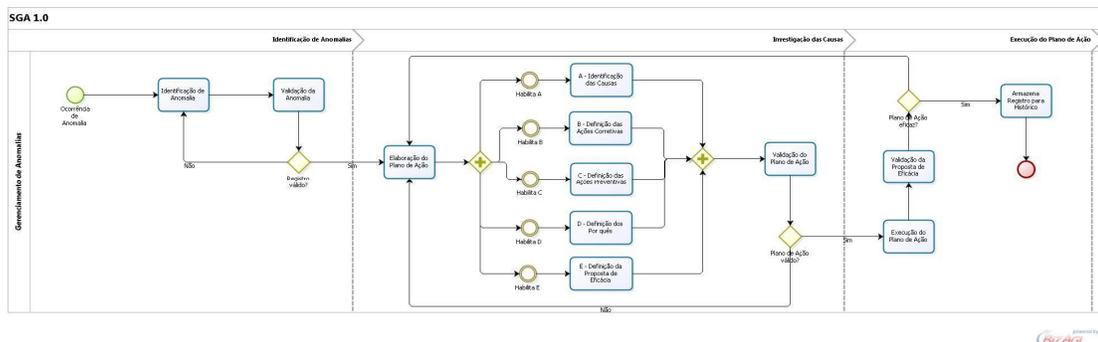


Figura A.1– Modelo do SGA no Bizagi

Após a modelagem do processo em BPMN deve-se especificar os papéis do processo na ferramenta. Para os dois PAISs foram utilizados os papéis: Analista e Operador do Processo. A figura A.2 ilustra os papéis do processo configurados na estrutura organizacional do Bizagi.

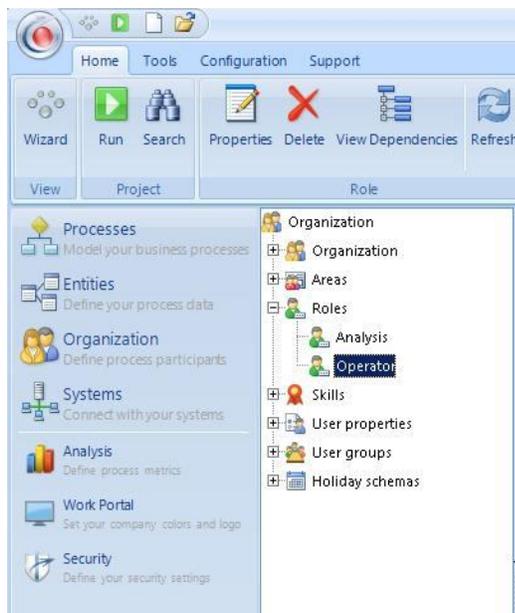
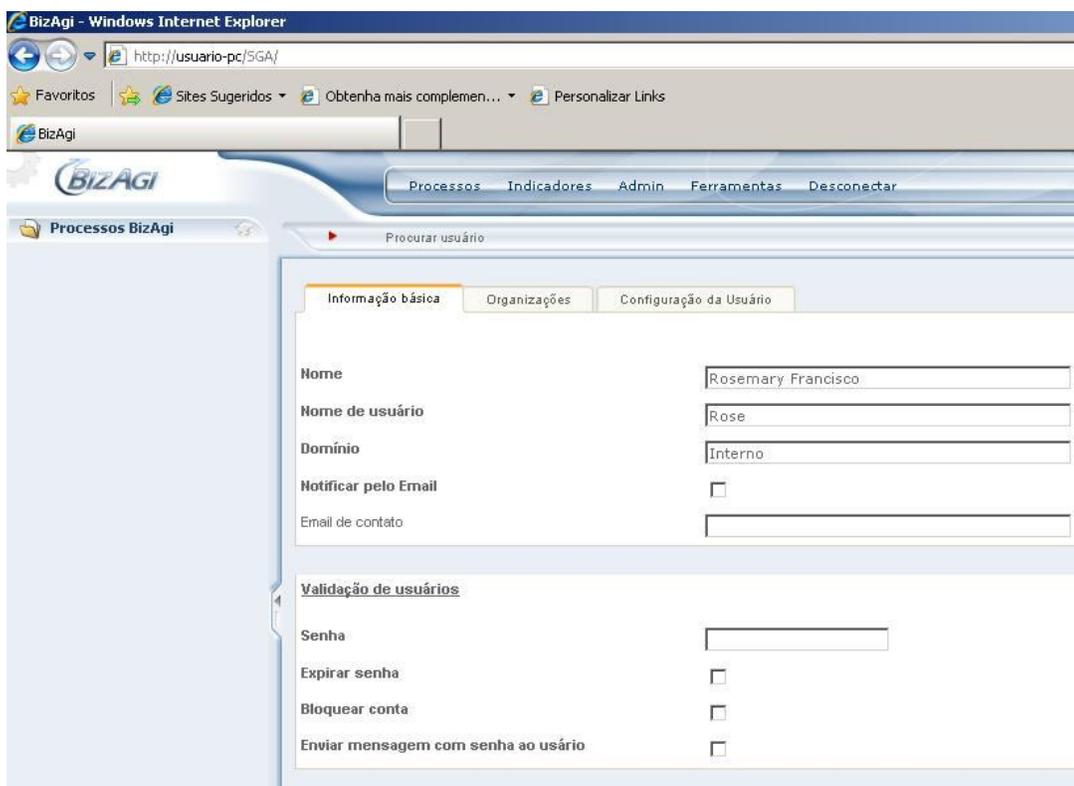


Figura A.2– Perfis do Processo no Bizagi

Para criar e configurar as permissões para os usuários do processo no Bizagi, toda a parte de configuração do processo deve estar finalizada e o processo já deve estar devidamente implementado. Este item, portanto, foi o último item configurado no Bizagi. As figuras A.3 e A.4 ilustram, respectivamente, a criação de um usuário no Bizagi e a atribuição deste usuário ao perfil de operador. Esta configuração deve ser feita na interface gráfica do Bizagi utilizada para a execução dos processos. Este tipo de configuração será feito por meio do item: Admin >> Usuários do menu superior da interface gráfica e será permitida somente para os usuários que possuem o perfil de administrador do processo configurado no Bizagi.



Informação básica	
Nome	Rosemary Francisco
Nome de usuário	Rose
Domínio	Interno
Notificar pelo Email	<input type="checkbox"/>
Email de contato	

Validação de usuários	
Senha	
Expirar senha	<input type="checkbox"/>
Bloquear conta	<input type="checkbox"/>
Enviar mensagem com senha ao usuário	<input type="checkbox"/>

Figura A.3– Criação de usuário no Bizagi

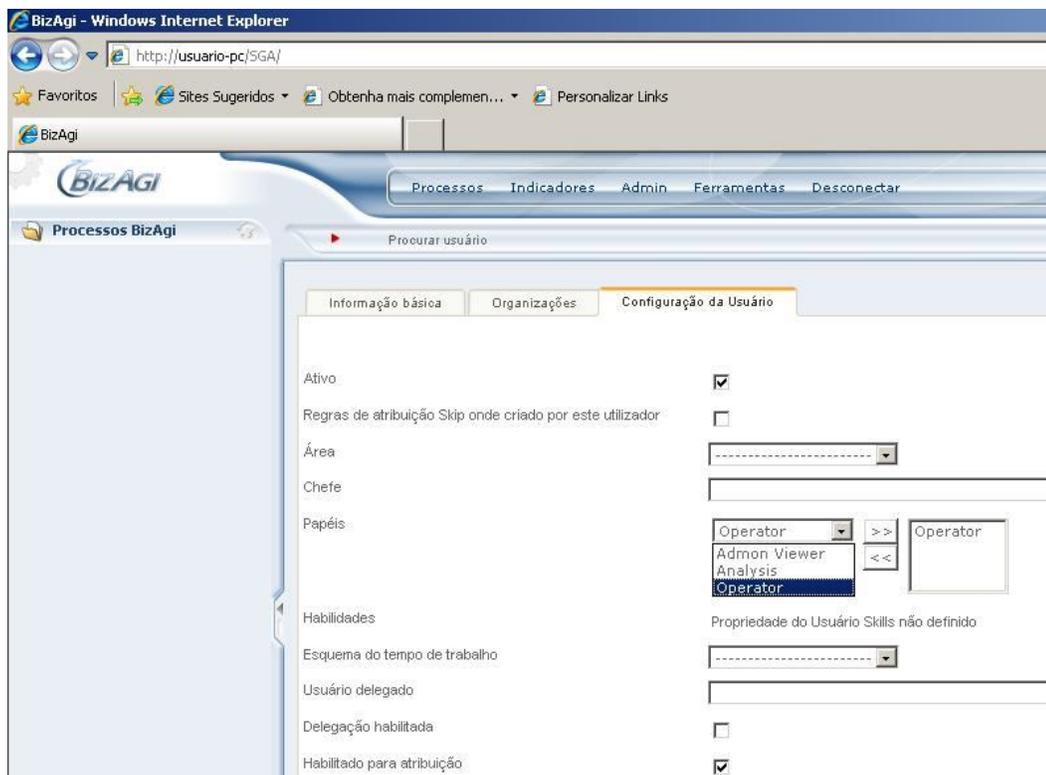


Figura A.4– Atribuição de perfil para o usuário no Bizagi

A configuração do WebService criada para possibilitar a troca de informações do Bizagi com o SCS é apresentada nas figuras A.5 a A.9. No Bizagi a configuração com o WebService foi criada por meio do recurso: *Expressions* disponível na área de configuração do processo. Este recurso permite a criação de uma expressão / rotina que poderá ser chamada em qualquer etapa do fluxo do processo. Para o presente trabalho, foi criada a expressão: “Consulta SCS” que faz a conexão com o WebService para enviar os dados de execução do processo para o SCS e obter o guia com a seqüência de execução das atividades. A figura A.5 ilustra a área de configuração do processo e a expressão: “Consulta SCS” criada.

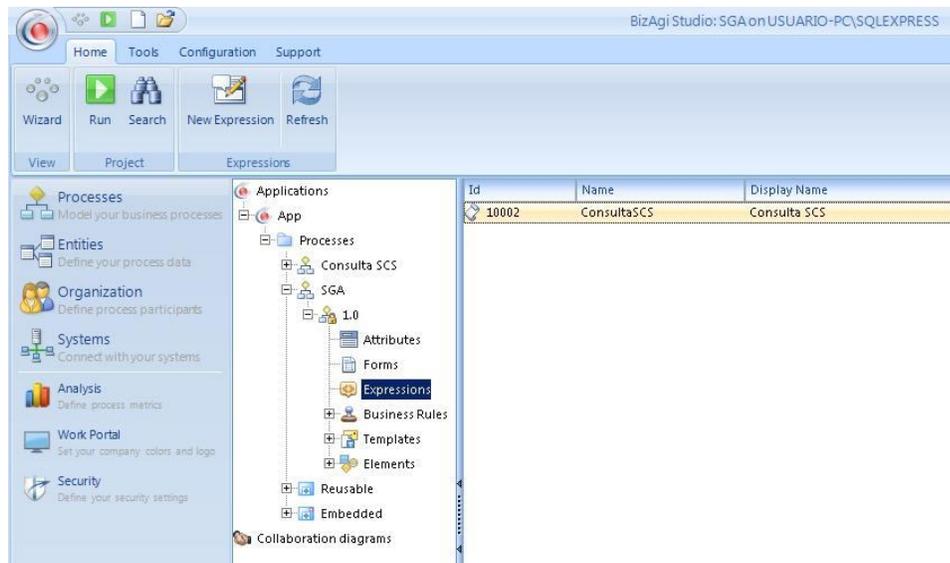


Figura A.5– Expressão Consulta SCS criada no Bizagi

A figura A.6 ilustra as propriedades da expressão: “Consulta SCS”. Por meio desta figura é possível verificar que a expressão criada refere-se a uma rotina de conexão com um Webservice.

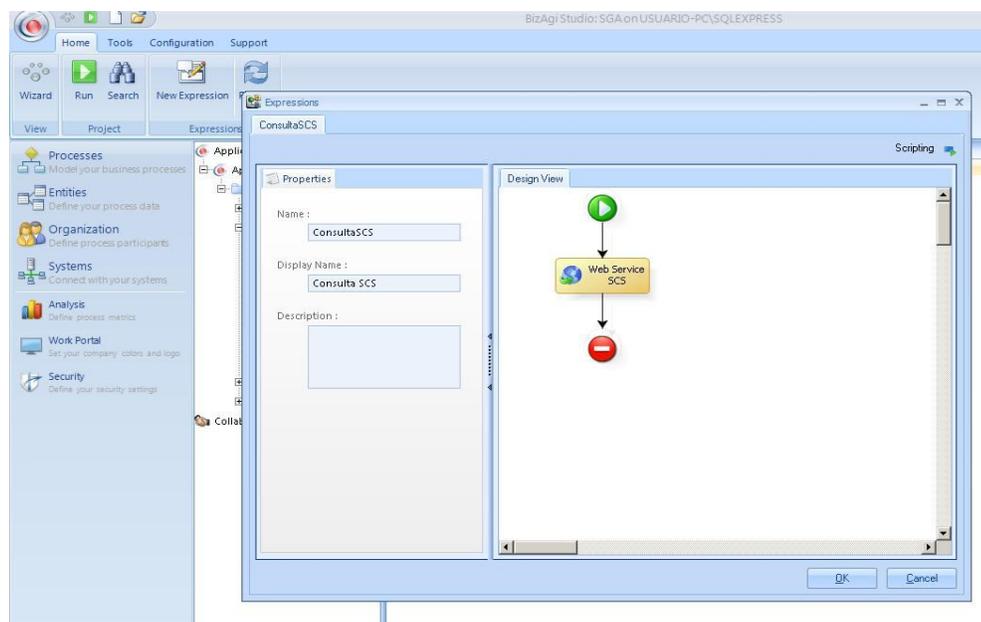


Figura A.6– Propriedades da Consulta SCS

A figura A.7 ilustra a configuração da conexão com o Webservice, por meio do caminho de acesso ao Webservice, localização do Webservice na rede interna ou mesmo externa. Para o exemplo ilustrado na figura A.7, o Webservice é

acessado por meio da rede interna, porém, nada impede que o acesso configurado seja externo à rede onde o processo de negócio é executado.

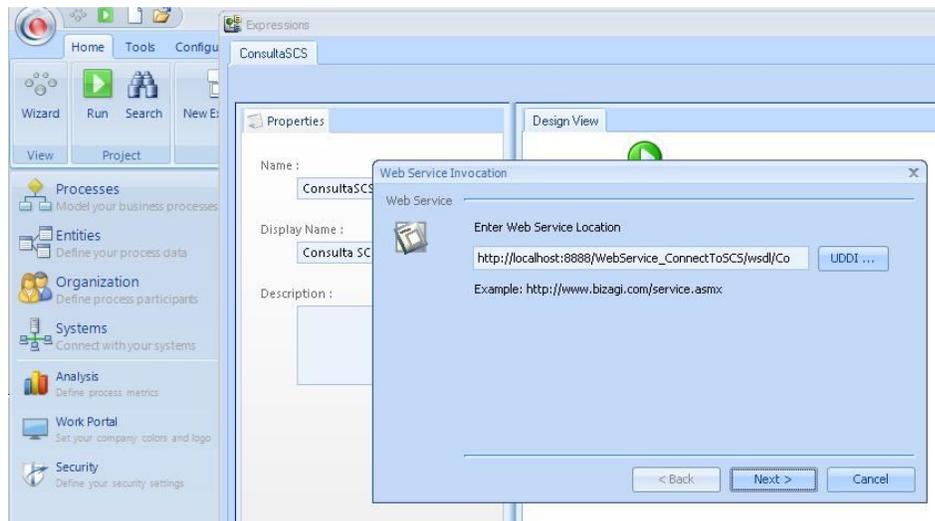


Figura A.7– Configuração da localização do Webservice no Bizagi

A figura A.8 ilustra a localização do Webservice pelo Bizagi e a apresentação dos métodos disponíveis neste Webservice para a configuração. O Webservice criado foi implementado apenas com um método, o método *readFromClient*. Este é o método que será utilizado na configuração do Webservice para envio dos dados da execução do processo. O método *readFromClient* recebe um único parâmetro do tipo string. Este parâmetro possuirá as informações acerca dos seguintes dados do processo: identificador da instância do processo (case), atividade executada e evento disparado, além dos dados do usuário: identificador, nome e perfil do usuário no processo.

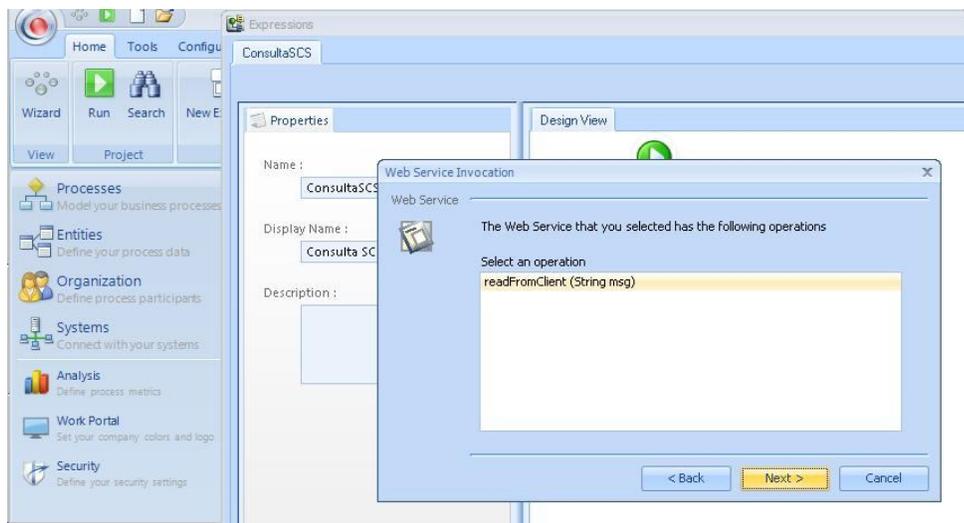


Figura A.8– Métodos do Webservice apresentados no Bizagi

A figura A.9, por sua vez, ilustra a configuração do valor do parâmetro que será encaminhado pelo Bizagi para o Webservice e, conseqüentemente, para o SCS. O valor do parâmetro será composto dos dados da execução do processo já mencionados.

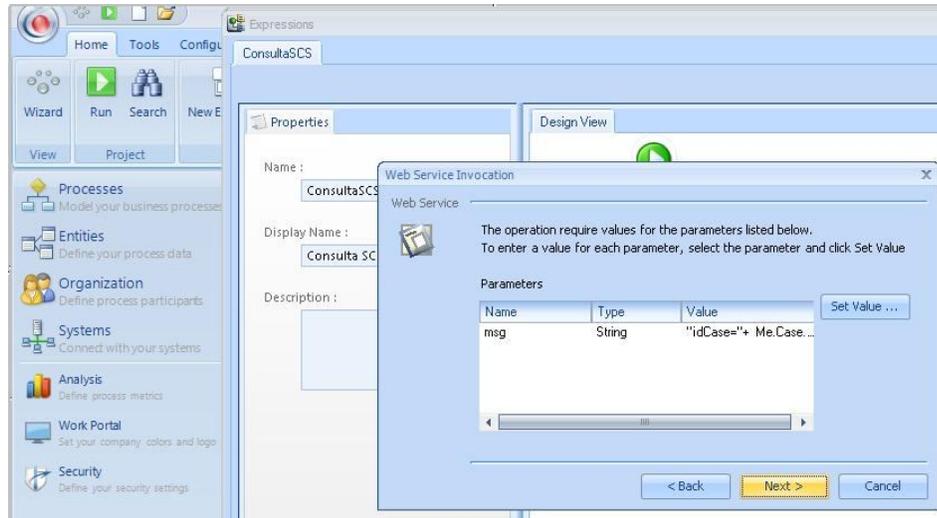


Figura A.9– Configuração do valor do parâmetro do método do Webservice no Bizagi

Após a configuração da expressão que permite a conexão com o Webservice, deve-se configurar os eventos do processo para permitir que a cada disparo de evento, o Webservice seja chamado e o SCS informado sobre o disparo do evento na execução do processo. A figura A.10 ilustra a tela do Bizagi que permite a configuração dos eventos no processo. Como ilustrado na figura, foram configurados os eventos de início e fim das atividades: A, B, C, D e E.

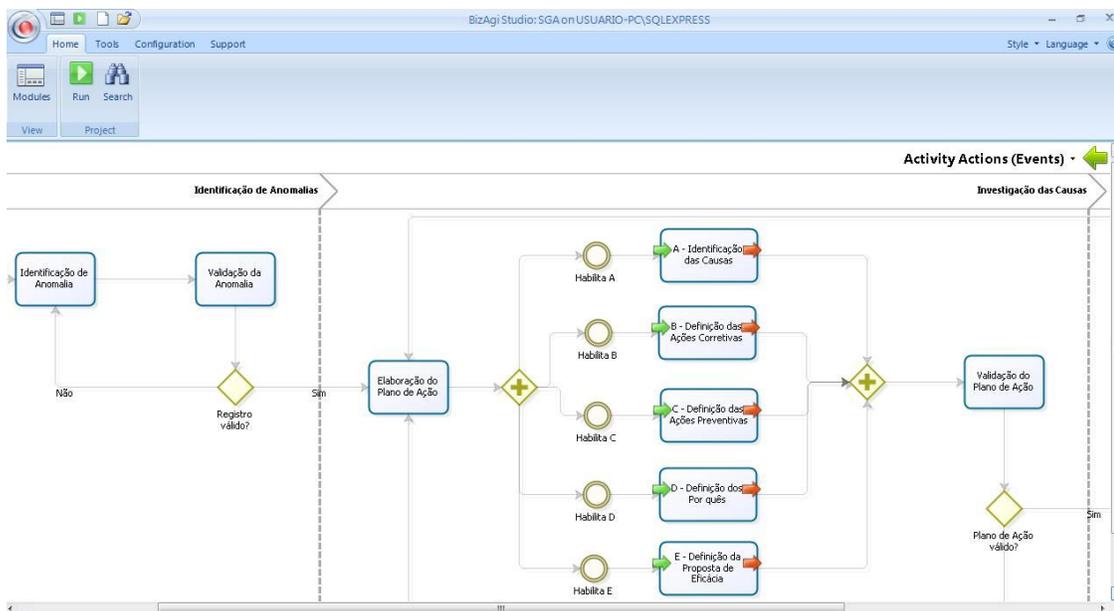


Figura A.10– Configuração dos eventos das atividades do processo no Bizagi

Por fim, a configuração do Widget é apresentada nas figuras A.11 a A.14. O Widget deve ser configurado no Bizagi como um tipo chamado: *UserField*. Este *UserField* é incluído em todas as telas do processo que representam a execução das atividades que são monitoradas pelo SCS. Na figura A.11 é apresentado o *UserField* que foi criado para permitir a apresentação do Widget do SCS nas telas de execução das atividades no Bizagi.

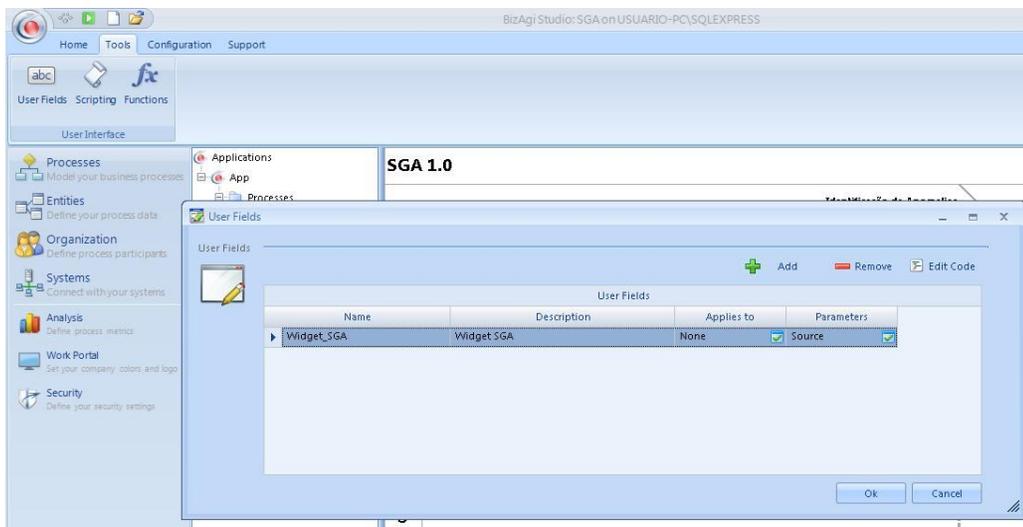


Figura A.11– UserField criado para apresentar o Widget no Bizagi

A figura A.12 ilustra o código que foi desenvolvido para o UserField apresentar o Widget do SCS. Como ilustrado na figura A.12, para que o Widget apresente as informações do guia do SCS em formato de semáforo adequadamente, com base no cenário em que se encontra, é necessário informar ao Widget qual é a instância do processo da atividade que está sendo executada, além da informação da atividade e o usuário executante.

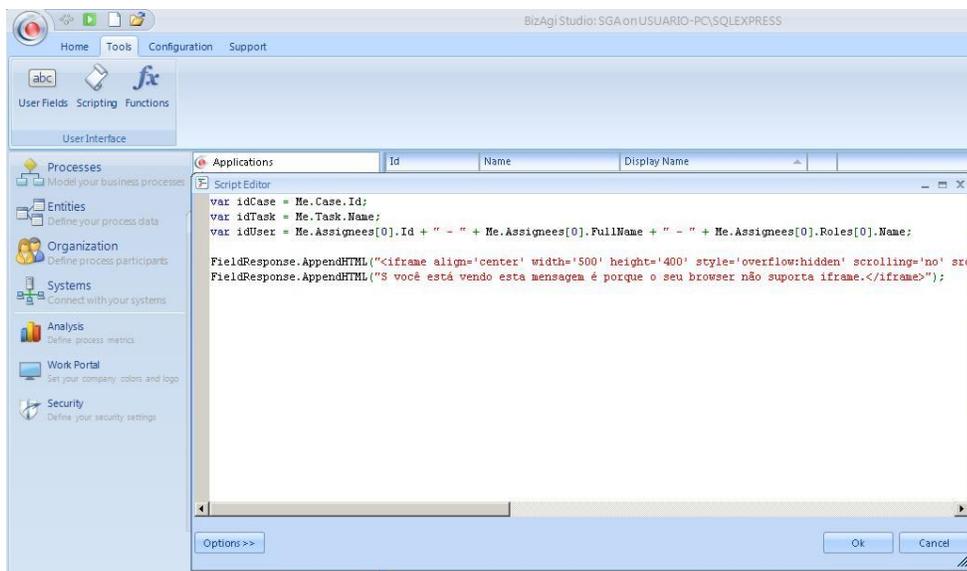


Figura A.12– Código utilizado no UserField para a configuração do Widget

A figura A.13 ilustra o formulário que foi criado para a execução das atividades no processo de negócio SGA. Neste formulário foi incluído *UserField* criado para ilustrar o guia de atividades retornado pelo SCS por meio do Widget.

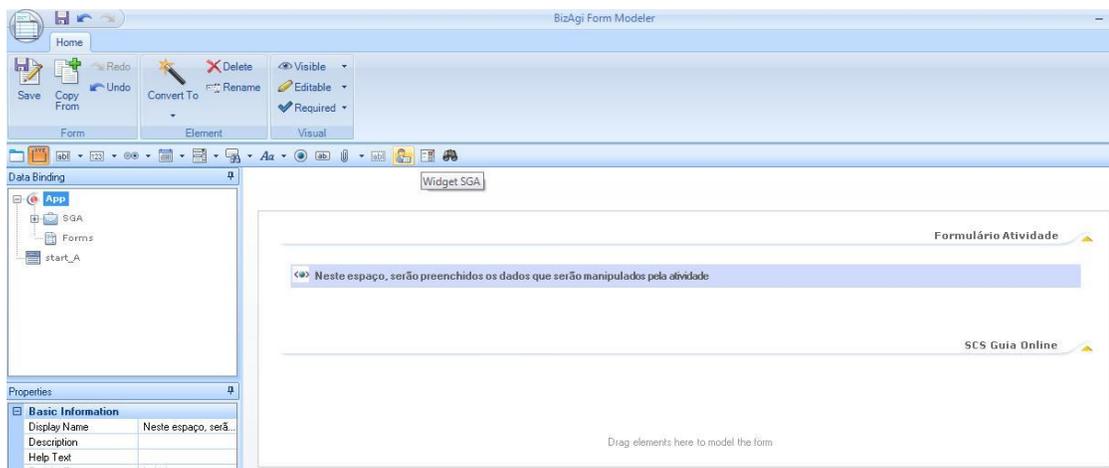


Figura A.13– Selecionado o UserField para a configuração do Widget no formulário das atividades

Finalmente, a figura A.14 ilustra o Widget configurado no formulário de atividades do processo.

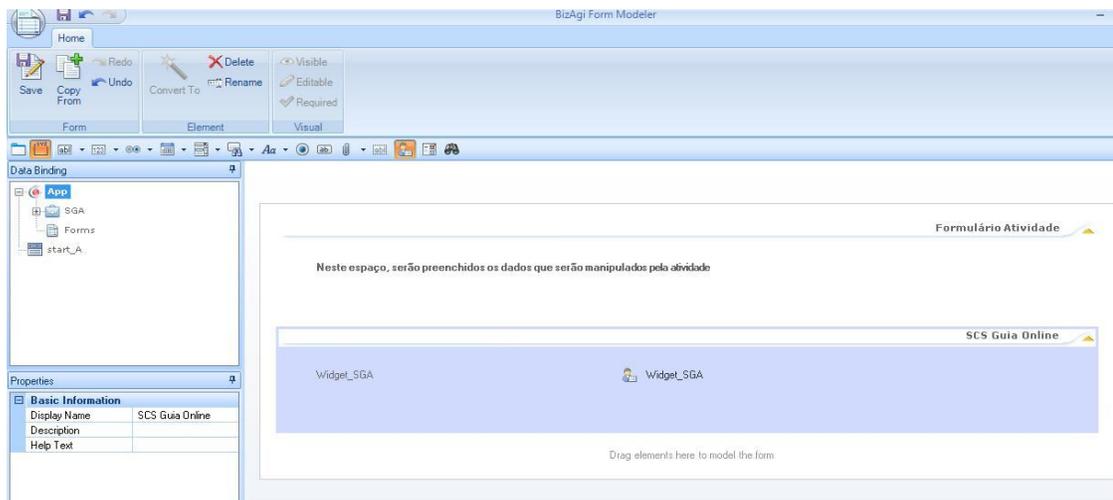


Figura A.14– UserField configurado no formulário das atividades

Após finalizar estas configurações no Bizagi, foi possível realizar o experimento e testar a execução do SGA com a supervisão do SCS.

APÊNDICE B – CONFIGURAÇÃO DO PAIS YAWL

Conforme ilustra a figura B.1, o modelo do processo de negócio SGA foi implementado no YAWL utilizando a sua notação própria. Uma das vantagens do Bizagi sob o YAWL é a interface mais intuitiva para a elaboração do modelo. Porém, o YAWL também possui outras vantagens que o Bizagi não possui, como a possibilidade de criar uma interface totalmente personalizada para a execução das atividades. A figura B.2 ilustra a configuração das atividades para utilizar uma interface personalizada.

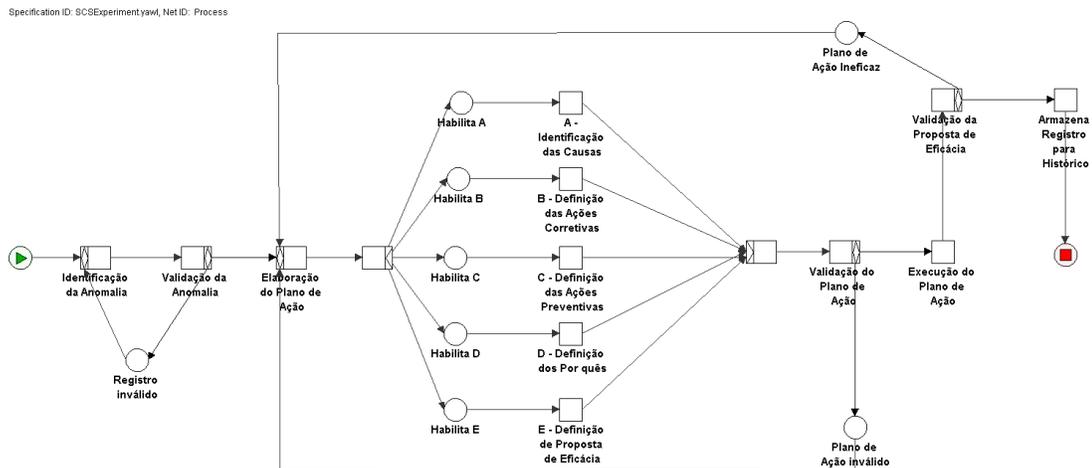


Figura B.15– Modelo do SGA no YAWL

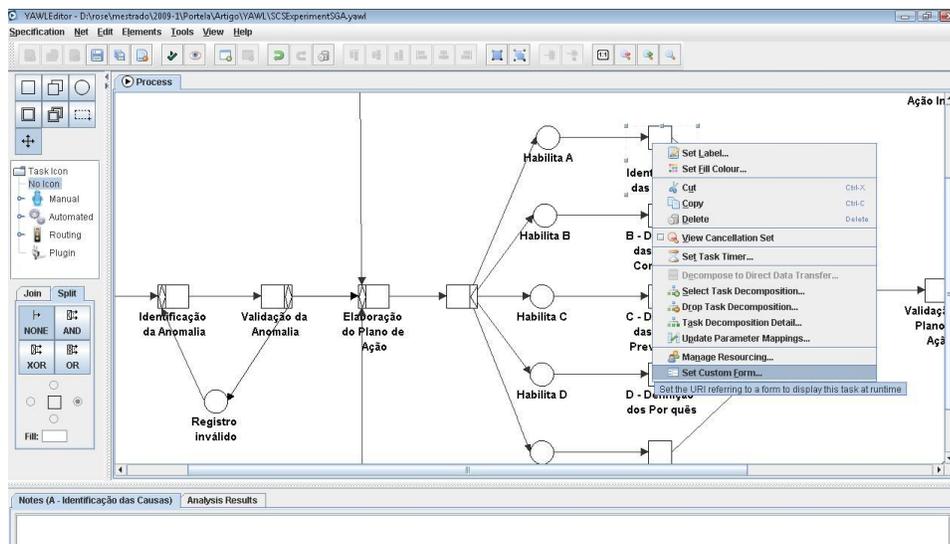


Figura B.16– Configurando a interface personalizada no YAWL

Após a modelagem do processo deve-se especificar os papéis do processo na ferramenta. Para os dois PAISs foram utilizados os papéis: Analista e Operador do Processo. A figura B.3 ilustra os papéis do processo configurados na estrutura organizacional do YAWL.

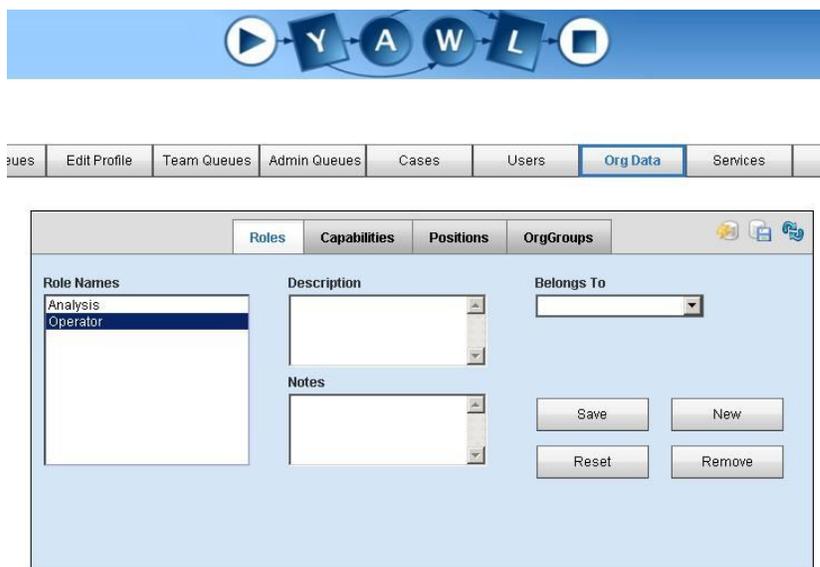


Figura B.17– Perfis do Processo no YAWL

Para criar e configurar os perfis e as permissões para os usuários do processo no YAWL, toda a parte de configuração do processo deve estar finalizada e o processo já deve estar devidamente implementado. A figura B.4 ilustra a criação de um usuário no YAWL e a atribuição deste usuário ao perfil de analista. Esta configuração deve ser feita na interface gráfica do YAWL utilizada para a execução dos processos. Este tipo de configuração será feito por meio do item: Users do menu superior da interface gráfica e será permitida somente para os usuários que possuem o perfil de administrador do processo configurado no YAWL.

Figura B.18– Criação de usuário no YAWL

A configuração do WebService criada para possibilitar a troca de informações do YAWL com o SCS é apresentada nas figuras B.5 e B.6. A figura B.5 ilustra a área de configuração do WebService e a figura B.6 ilustra o WebService já configurado na lista de serviços do YAWL.

Service Name	URI	Description
invokerService	http://localhost:8080/yaw/WSInvoker	Web Service Invoker Service
workletService	http://localhost:8080/workletService/ib	Worklet Dynamic Process Selection and Exception Service

Figura B.19– Configuração do WebService no YAWL

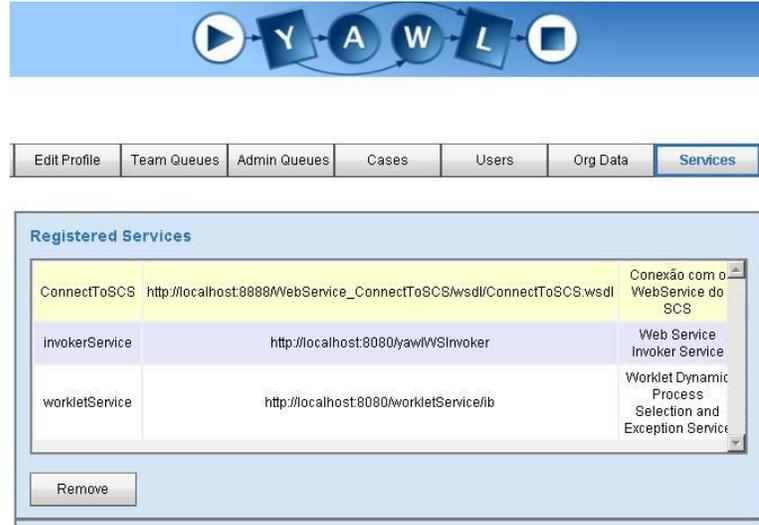


Figura B.20– WebService configurado no YAWL

Como foi utilizada uma interface personalizada para a execução das atividades no YAWL, diferente do Bizagi, toda a configuração dos eventos foi realizada por meio de código fonte em Java para WEB, que é o formato utilizado para a criação de interfaces gráficas no YAWL. O Widget também foi configurado direto nesta interface, não necessitando também uma configuração específica por meio da administração do YAWL. Após finalizar estas configurações no YAWL, foi possível realizar o experimento e testar a execução do SGA com a supervisão do SCS.

APÊNDICE C – IMPLEMENTAÇÃO SUPERVISORYCONTROLPROVIDER

A figura C.1 ilustra a implementação da classe SupervisoryControlProvider.

```

package org.processmining.plugins.operationalsupport.providers.scs;

import org.deckfour.xes.extension.std.XExtendedEvent;
import org.deckfour.xes.extension.std.XLifecycleExtension;
import org.deckfour.xes.extension.std.XLifecycleExtension.StandardModel;
import org.deckfour.xes.model.XEvent;
import org.deckfour.xes.model.XTrace;
import org.processmining.framework.plugin.PluginContext;
import org.processmining.framework.plugin.annotations.Plugin;
import org.processmining.framework.plugin.events.Logger.MessageLevel;
import org.processmining.framework.util.socket.ServiceEnvironment;
import org.processmining.models.operationalsupport.ProviderResponse;
import org.processmining.models.operationalsupport.Recommendation;
import org.processmining.models.operationalsupport.Request;
import org.processmining.models.operationalsupport.net.provider.Provider;
import org.processmining.models.operationalsupport.net.provider.impl.AbstractProvider;
import org.processmining.models.operationalsupport.net.service.OSService;
import org.processmining.plugins.operationalsupport.providers.scs.cmlevel.SupervisoryCMBusiness;

public class SupervisoryControlProvider extends AbstractProvider {
    ...
    public void populateResponse(ServiceEnvironment environment, Request request, ProviderResponse
response){
        environment.log("Supervisory Control Provider is processing request" + request.getId());
        ...
    }

    @Plugin(name = "SCS Provider", parameterLabels = {"Operational Support Service"}, returnLabels = {"SCS
Provider"}, returnTypes = { AbstractProvider.class }, userAccessible = true)
    public static Provider registerServiceProvider(final PluginContext context, OSService service){
        SupervisoryControlProvider scs = new SupervisoryControlProvider(service, context);
        context.getFutureResult(0).setLabel(context.getFutureResult(0).getLabel() + " on ports" + service);
        return scs;
    }
    ...
}

```

Figura C.21– Implementação da Classe SupervisoryControlProvider

APÊNDICE D – IMPLEMENTAÇÃO DO WEBSERVICE

A figura D.1 ilustra o WSDL do Webservice implementado para conexão do PAIS com o SCS.

```

<?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions targetNamespace="http://connection" xmlns:apachesoap="http://xml.apache.org/xml-soap" xmlns:impl="http://connection"
  xmlns:intf="http://connection" xmlns:wscdl="http://schemas.xmlsoap.org/wsdl/" xmlns:wscdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
- <!--
  WSDL created by Apache Axis version: 1.4
  Built on Apr 22, 2006 (06:55:48 PDT)
-->
- <wsdl:types>
- <schema elementFormDefault="qualified" targetNamespace="http://connection" xmlns="http://www.w3.org/2001/XMLSchema">
  - <element name="readFromClient">
    - <complexType>
      - <sequence>
        <element name="msg" type="xsd:string" />
      </sequence>
    </complexType>
  </element>
  - <element name="readFromClientResponse">
    - <complexType>
      - <sequence>
        <element name="readFromClientReturn" type="xsd:string" />
      </sequence>
    </complexType>
  </element>
</schema>
</wsdl:types>
+ <wsdl:message name="readFromClientRequest">
+ <wsdl:message name="readFromClientResponse">
+ <wsdl:portType name="ConnectToSCS">
+ <wsdl:binding name="ConnectToSCSSoapBinding" type="impl:ConnectToSCS">
- <wsdl:service name="ConnectToSCSService">
  - <wsdl:port binding="impl:ConnectToSCSSoapBinding" name="ConnectToSCS">
    <wsdlsoap:address location="http://localhost:8888/WebService_ConnectToSCS/services/ConnectToSCS" />
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

Figura D.22– WSDL do Webservice

APÊNDICE E – ESPECIFICAÇÃO DO PLANO E CASOS DE TESTE

A tabela E.1 apresenta a especificação definida para o plano de testes para a condução do experimento com o SCS.

Tabela E.1–Plano de testes elaborado para a condução do experimento

Escopo do Plano	Executar o processo de negócio SGA nos PAISs com a supervisão do SCS a fim de validar se as regras de negócio definidas para o processo estão sendo observadas e garantidas pelo SCS.
Artefatos de Entrada	Modelo do Processo e Regras de negócio definidas no capítulo 4, além das configurações do SCS e PAISs.
Itens do Teste	Os itens que serão avaliados no testes são : <ul style="list-style-type: none"> – Cenário do processo – Diagramas de seqüência elaborados para demonstrar a seqüência de atividades percorrida pelo SCS durante a supervisão do processo, vide capítulo 4.
Tipo do Teste	Os tipos de testes que serão realizados são: <ul style="list-style-type: none"> – Funcionalidade => executa as funções associadas aos diagramas de seqüência definidos garantindo que: (1) todos os cenários (fluxos alternativos e secundários) sejam percorridos; (2) as entradas fornecidas (válidas e inválidas) resultam em comportamentos definidos na especificação. Objetivo: Garantir a corretude das funcionalidades descritas a partir dos diagramas de seqüência. Estes testes devem garantir que a aplicação se comporte conforme os fluxos de eventos e diagramas elaborados; – Interface => testa o comportamento do Widget e sua integração com o PAIS. Objetivo: Assegurar que o comportamento, requisitos e projeto gráfico definidos para a interface sejam atendidos; – Configuração => assegura que as configurações especificadas atendem às necessidades para o perfeito funcionamento dos sistemas; – Ciclo de Negócio => garante que os sistemas funcionem apropriadamente durante um ciclo de atividades relativas ao modelo do processo de negócio e que ao final desse ciclo todos os resultados esperados foram obtidos.
Abordagem do Teste	As abordagens de testes que serão utilizadas são : <ul style="list-style-type: none"> – Teste Caixa Preta => este tipo de teste analisa o item a ser testado sob o aspecto funcional. O foco dos testes de caixa preta está nas entradas, saídas esperadas e saídas obtidas, independentemente da implementação do item a ser testado. – Teste Integrado => este tipo de teste verifica a compatibilidade da interface entre os componentes da aplicação, constituindo uma técnica sistemática para a construção da estrutura do programa. O objetivo deste teste é garantir, em uma maior escala, que as rotinas implementadas estejam interagindo entre si conforme definido; – Teste de Sistema => este tipo de teste verifica se a aplicação está funcionando como um todo. A integração da aplicação com o ambiente operacional similar ao de produção (hardware, pessoas e outras aplicações) é testada. O objetivo deste teste é verificar uma combinação de requisitos funcionais instanciados sob diferentes condições.
Ambiente do Teste	O ambiente utilizado para a realização dos testes é descrita abaixo : <ul style="list-style-type: none"> – Sistema Operacional Windows Vista – versão Ultimate

	<ul style="list-style-type: none"> - Eclipse Java EE IDE for Web Developers – versão GALILEO - ProM – versão 5.1 - Bizagi Xpress Edition – versão 9.1.4.1002 - YAWL 4 Study – versão 2.0.1
--	--

As tabelas E.2, E.3 e E.4 apresentam a especificação definida para os casos de testes para a condução do experimento com o SCS.

Tabela E.2–Caso de testes 01 elaborado para a condução do experimento

Identificador do Teste	Caso de testes 01	
Cenário do Teste	Executar a sequencia de atividades: A, E, B, C, D	
Objetivo do Teste	Validar se as regras de controle estão sendo corretamente aplicadas pelo SCS	
Ambiente necessário	Ambiente especificado no plano de testes	
Pré-requisitos	Configurações no SCS e PAISs devidamente realizadas conforme detalhado nos tópicos deste capítulo.	
Execução do Caso de Testes		
Procedimento	Entradas	Resultado Esperado
Iniciar a execução do processo		O guia do SCS deve apresentar a atividade : C como atividade inabilitada para execução.
Iniciar a execução da atividade : A – Identificação das Causas	<ul style="list-style-type: none"> - Atividade: A - Evento: start_A - Dados do usuário 	O guia do SCS deve apresentar a atividade : C como atividade inabilitada para execução.
Finalizar a execução da atividade : A – Identificação das Causas	<ul style="list-style-type: none"> - Atividade: A - Evento: complete_A - Dados do usuário 	O guia do SCS deve apresentar a atividade : C como atividade inabilitada para execução.
Iniciar a execução da atividade : E – Definição de Proposta de Eficácia	<ul style="list-style-type: none"> - Atividade: E - Evento: start_E - Dados do usuário 	O guia do SCS deve apresentar a atividade : C como atividade inabilitada para execução.
Finalizar a execução da atividade : E – Definição de Proposta de Eficácia	<ul style="list-style-type: none"> - Atividade: E - Evento: complete_E - Dados do usuário 	O guia do SCS deve apresentar a atividade : C como atividade inabilitada para execução.
Iniciar a execução da atividade : B – Definição das Ações Corretivas	<ul style="list-style-type: none"> - Atividade: B - Evento: start_B - Dados do usuário 	O guia do SCS deve apresentar as atividades : B,C e D como atividades inabilitadas para execução.
Finalizar a execução da atividade : B – Definição das Ações Corretivas	<ul style="list-style-type: none"> - Atividade: B - Evento: complete_B - Dados do usuário 	O guia do SCS deve apresentar a atividade : B como atividade inabilitada para execução.
Iniciar a execução da atividade : C – Definição das Ações Preventivas	<ul style="list-style-type: none"> - Atividade: C - Evento: start_C - Dados do usuário 	O guia do SCS deve apresentar a atividade : C como atividade inabilitada para execução.
Finalizar a execução da atividade : C – Definição das Ações Preventivas	<ul style="list-style-type: none"> - Atividade: C - Evento: complete_C - Dados do usuário 	O guia do SCS deve apresentar a atividade : C como atividade inabilitada para execução.
Iniciar a execução da atividade : D – Definição dos Por quês	<ul style="list-style-type: none"> - Atividade: D - Evento: start_D - Dados do usuário 	O guia do SCS deve apresentar as atividades : B,C e D como atividades inabilitadas para execução.
Finalizar a execução da atividade : D – Definição dos Por quês	<ul style="list-style-type: none"> - Atividade: D - Evento: complete_D - Dados do usuário 	O guia do SCS deve apresentar a atividade : C como atividade inabilitada para execução.

Tabela E.3–Caso de testes 02 elaborado para a condução do experimento

Identificador do Teste	Caso de testes 02	
 Cenário do Teste	Executar a sequencia de atividades: B, D, C, A, E	
Objetivo do Teste	Validar se as regras de controle estão sendo corretamente aplicadas pelo SCS	
Ambiente necessário	Ambiente especificado no plano de testes	
Pré-requisitos	Configurações no SCS e PAISs devidamente realizadas conforme detalhado nos tópicos deste capítulo.	
Execução do Caso de Testes		
Procedimento	Entradas	Resultado Esperado
Iniciar a execução do processo		O guia do SCS deve apresentar a atividade : C como atividade inabilitada para execução.
Iniciar a execução da atividade : B – Definição das Ações Corretivas	<ul style="list-style-type: none"> – Atividade: B – Evento: start_B – Dados do usuário 	O guia do SCS deve apresentar as atividades : B,C e D como atividades inabilitadas para execução.
Finalizar a execução da atividade : B – Definição das Ações Corretivas	<ul style="list-style-type: none"> – Atividade: B – Evento: complete_B – Dados do usuário 	O guia do SCS deve apresentar a atividade : B como atividade inabilitada para execução.
Iniciar a execução da atividade : D – Definição dos Por quês	<ul style="list-style-type: none"> – Atividade: D – Evento: start_D – Dados do usuário 	O guia do SCS deve apresentar as atividades : B,C e D como atividades inabilitadas para execução.
Finalizar a execução da atividade : D – Definição dos Por quês	<ul style="list-style-type: none"> – Atividade: D – Evento: complete_D – Dados do usuário 	O guia do SCS deve apresentar a atividade : C como atividade inabilitada para execução.
Iniciar a execução da atividade : C – Definição das Ações Preventivas	<ul style="list-style-type: none"> – Atividade: C – Evento: start_C – Dados do usuário 	O guia do SCS deve apresentar a atividade : C como atividade inabilitada para execução.
Finalizar a execução da atividade : C – Definição das Ações Preventivas	<ul style="list-style-type: none"> – Atividade: C – Evento: complete_C – Dados do usuário 	O guia do SCS deve apresentar a atividade : C como atividade inabilitada para execução.
Iniciar a execução da atividade : A – Identificação das Causas	<ul style="list-style-type: none"> – Atividade: A – Evento: start_A – Dados do usuário 	O guia do SCS deve apresentar a atividade : C como atividade inabilitada para execução.
Finalizar a execução da atividade : A – Identificação das Causas	<ul style="list-style-type: none"> – Atividade: A – Evento: complete_A – Dados do usuário 	O guia do SCS deve apresentar a atividade : C como atividade inabilitada para execução.
Iniciar a execução da atividade : E – Definição de Proposta de Eficácia	<ul style="list-style-type: none"> – Atividade: E – Evento: start_E – Dados do usuário 	O guia do SCS deve apresentar a atividade : C como atividade inabilitada para execução.
Finalizar a execução da atividade : E – Definição de Proposta de Eficácia	<ul style="list-style-type: none"> – Atividade: E – Evento: complete_E – Dados do usuário 	O guia do SCS deve apresentar a atividade : C como atividade inabilitada para execução.

Tabela E.4–Caso de testes 03 elaborado para a condução do experimento

Identificador do Teste	Caso de testes 03	
 Cenário do Teste	Executar a sequencia de atividades: E, D, A, B, C	
Objetivo do Teste	Validar se as regras de controle estão sendo corretamente aplicadas pelo SCS	
Ambiente necessário	Ambiente especificado no plano de testes	
Pré-requisitos	Configurações no SCS e PAISs devidamente realizadas conforme detalhado nos tópicos deste capítulo.	
Execução do Caso de Testes		
Procedimento	Entradas	Resultado Esperado
Iniciar a execução do processo		O guia do SCS deve apresentar a atividade : C como atividade inabilitada para execução.

Iniciar a execução da atividade : E – Definição de Proposta de Eficácia	<ul style="list-style-type: none"> – Atividade: E – Evento: start_E – Dados do usuário 	O guia do SCS deve apresentar a atividade : C como atividade inabilitada para execução.
Finalizar a execução da atividade : E – Definição de Proposta de Eficácia	<ul style="list-style-type: none"> – Atividade: E – Evento: complete_E – Dados do usuário 	O guia do SCS deve apresentar a atividade : C como atividade inabilitada para execução.
Iniciar a execução da atividade : D – Definição dos Por quês	<ul style="list-style-type: none"> – Atividade: D – Evento: start_D – Dados do usuário 	O guia do SCS deve apresentar as atividades : B,C e D como atividades inabilitadas para execução.
Finalizar a execução da atividade : D – Definição dos Por quês	<ul style="list-style-type: none"> – Atividade: D – Evento: complete_D – Dados do usuário 	O guia do SCS deve apresentar a atividade : C como atividade inabilitada para execução.
Iniciar a execução da atividade : A – Identificação das Causas	<ul style="list-style-type: none"> – Atividade: A – Evento: start_A – Dados do usuário 	O guia do SCS deve apresentar a atividade : C como atividade inabilitada para execução.
Finalizar a execução da atividade : A – Identificação das Causas	<ul style="list-style-type: none"> – Atividade: A – Evento: complete_A – Dados do usuário 	O guia do SCS deve apresentar a atividade : C como atividade inabilitada para execução.
Iniciar a execução da atividade : B – Definição das Ações Corretivas	<ul style="list-style-type: none"> – Atividade: B – Evento: start_B – Dados do usuário 	O guia do SCS deve apresentar as atividades : B,C e D como atividades inabilitadas para execução.
Finalizar a execução da atividade : B – Definição das Ações Corretivas	<ul style="list-style-type: none"> – Atividade: B – Evento: complete_B – Dados do usuário 	O guia do SCS deve apresentar a atividade : B como atividade inabilitada para execução.
Iniciar a execução da atividade : C – Definição das Ações Preventivas	<ul style="list-style-type: none"> – Atividade: C – Evento: start_C – Dados do usuário 	O guia do SCS deve apresentar a atividade : C como atividade inabilitada para execução.
Finalizar a execução da atividade : C – Definição das Ações Preventivas	<ul style="list-style-type: none"> – Atividade: C – Evento: complete_C – Dados do usuário 	O guia do SCS deve apresentar a atividade : C como atividade inabilitada para execução.